Computer Programs for Amateur Radio

Wayne Overbeck and James A. Steffen

High-performance programs written in BASIC for your Apple™, TRS-80™, or Commodore 64™. Plus Conversion Guides for the IBM PC™

COMPUTER PROGRAMS FOR AMATEUR RADIO

Wayne Overbeck, N6NB
James A. Steffen, KC6A

HAYDEN BOOK COMPANY
a division of Hayden Publishing Company, Inc.
Hasbrouck Heights, New Jersey

Acquisitions Editor: PRIJONO HARDJOWIROGO
Production Editor: MAUREEN CONNELLY
Art Director: JIM BERNARD
Original artwork: JOHN McAUSLAND
Text design: CHERYL HUDSON
Cover design: JOHN M-RÖBLIN
Cover photography: RON SCALERA
Cover equipment courtesy of KJI Electronics (Cedar Grove, NJ)
Composition: McFARLAND GRAPHICS AND DESIGN
Printed and bound by: WICKERSHAM PRINTING COMPANY, INC.

Copyright © 1984 by HAYDEN BOOK COMPANY. All rights reserved. No part of this book may be reprinted, or reproduced, or utilized in any form or by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying and recording, or in any information storage and retrieval system, without permission in writing from the Publisher.

Printed in the United States of America

3 4 5 6 7 8 9 PRINTING

85 86 87 88 89 90 91 92 YEAR

PREFACE

Not long ago, a feature article in a national ham radio magazine dismissed personal computers as terrific for home video games and Morse/RTTY applications, but not much good for anything else that most radio amateurs would be interested in.

Granted, computers are good for Morse and RTTY, and the games are fun. However, computers will also do a lot of other very useful things in the hamshack. For example, amateur radio is a data-intensive hobby. Many ham radio activities require record-keeping or mathematical calculations—or both. There are detailed records to be maintained if you are working toward five-band DXCC, VUCC, WAS—or any of many other amateur radio awards. If you are interested in DXing, the rising and setting of the sun—at your home and at places half a world away—is very relevant. And it's also important to know the distance and beam heading from your station to many other cities or countries. Also, even at a time when the government regulations requiring logkeeping have been relaxed, a well-organized log is probably important for many of the things you do.

If your interests include contest operating, you have an enormous amount of data to process quickly—and a computer can do it far more rapidly and accurately than you can by hand. If you're interested in VHF operating, keeping track of grid locators is becoming increasingly important. If moonbounce is your

thing, you need to track the moon across the sky, even when you can't see it because of cloud cover.

And whatever else interests you about ham radio, you almost certainly have some interest in antennas. You want to know if your antennas work as well as they should, and you probably have some interest in building even better ones.

Yes, a computer can decipher RTTY and Morse—and it can play video games—but it can also do many other things very well.

The programs in this book range from relatively short ones that perform mathematical computations to sophisticated contest operating and data base management programs for the operator who takes ham radio and computers seriously. Most of the programs are new ones that have never been published anywhere before. They are the result of many hours of programming time, based on the authors' combined 50 years of experience in all phases of amateur radio from DXing, contests, and moonbounce to just plain weekend rag-chewing.

In selecting programs to include in this book, we chose some that will run on even the simplest home computers and others that require a disk drive and printer. Also, we tried to avoid needless duplication of software that is readily available elsewhere—Morse/RTTY packages, satellite-tracking programs (which are available directly from AMSAT), and engineering programs like those in another Hayden book, Jules Gilder's *Basic Computer Programs for Science and Engineering*.

Rather, we emphasized programs that do for radio amateurs what computers have traditionally done best: *data processing*. Our goal was to prepare a package of BASIC programs that would offer a new level of performance to meet the data-handling needs of radio amateurs.

The programs presented here have been fully tested and proven on several popular brands of personal computers. They were written especially for Apple II, Commodore 64, and computers that run Microsoft BASIC, such as the TRS-80, IMB PC and CP/M-based systems. There are separate listings of almost every program for Apple, Commodore, and TRS-80, with specific instructions for the very few changes needed to make the TRS-80 versions run on an IBM PC or CP/M systems. The text even shows how some of the programs can be adapted for the low-cost Timex-Sinclair 1000 (ZX-81) and Commodore VIC-20.

Moreover, there is a chapter describing the major differences in programming techniques required for various brands of personal computers. And for those who are contemplating the purchase of their first computer, there are introductory chapters explaining the fundamentals of personal computing—based on the authors' professional and at-home experience with large and small computers. Among the many computers now on the market, there are enormous differences in flexibility, power, and processing speed, among other features. This book includes a no-holds-barred discussion of the virtues and foibles of various brands.

Computers and amateur radio. Rarely has there been a better marriage of two hobbies than this one. Many radio amateurs are technically oriented experimenters, and they're not the least bit uncomfortable in a world of chips and

transistors, cables and power supplies. They're used to seeing boxes lined up on a table with wires running around them. When personal computers came along, thousands of radio amateurs happily added personal computers to their ham shacks, and quickly found themselves caught up in a brand new technical hobby. If you're one of these people, we hope you enjoy the programs in this book.

However, while thousands of radio amateurs count personal computing among their hobbies, very few would list *typing* and *proofreading* as hobbies. Therefore, almost all of the programs in this book are available from the publisher on a floppy disk ready to run on the Apple II. Given sufficient interest, the ready-to-run version of this software will also be offered in other popular disk formats.

Wayne Overbeck, N6NB
James Steffen, KC6A

CONTENTS

1	A Gallery of Programs for Radio Amateurs	1
2	The Evolution of the Microcomputer	9
3	Microcomputer Hardware	21
4	Things Your Mother (and Maybe Your Dealer) Never Told You about Microcomputers	40
5	BASIC: One Language, Many Dialects	54
6	Data Base Management Programs for Amateur Radio	76
7	Latitude-Longitude Programs for DX Work	123
8	Programs for the Contest Operator	170
9	Programs for Antenna Design and Evaluation	271
10	To the Moon—and Beyond	294
	In Case of Difficulty	325

LIST OF PROGRAMS

Note: All TRS-80 and CP/M (Microsoft BASIC) listings are also intended for use on IBM PC, with minor modifications which are described on pages 72-74. An asterisk ("*") indicates a program can be set up for an unexpanded VIC-20 (see pages 61 and 68 for details).

A DEMONSTRATION PROGRAM

Mini-Logger (TRS-80 and CP/M), *58*
Mini-Logger (Apple), *59*
Mini-Logger (Commodore*), *60*

DATA BASE MANAGEMENT PROGRAMS

Logbook (TRS-80 and CP/M), *82*
Logbook (Apple), *87*
Logbook (Commodore), *91*
Radio Awards Data Base (TRS-80 and CP/M), *101*
Radio Awards Data Base (Apple), *106*
Radio Awards Data Base (Commodore), *112*
Setup Program for Awards Data Base (TRS-80 and CP/M), *118*
Setup Program for Awards Data Base (Apple), *119*
Setup Program for Awards Data Base (Commodore), *119*
Gridlocator (standard version*), *121*

LATITUDE-LONGITUDE BASED PROGRAMS

Latitude-Longitude Data File (Standard version), *126*
Beamheading Chart (TRS-80 and CP/M), *134*
Beamheading Chart (Apple), *136*
Beamheading Chart (Commodore), *137*
DX Display (TRS-80 and CP/M), *140*
DX Display (Apple), *142*
DX Display (Commodore), *144*
Sunrise Chart (TRS-80 and CP/M), *146*
Sunrise Chart (Apple), *147*
Sunrise Chart (Commodore), *149*
Grayline (TRS-80 and CP/M), *153*
Grayline (Apple), *157*
Grayline (Commodore), *160*
Sunrises Anywhere (TRS-80 and CP/M), *164*
Sunrises Anywhere (Apple), *165*
DX Checker (standard version*), *168*

CONTEST LOGGING AND DUPECHECKING PROGRAMS

Dupechecker (TRS-80 and CP/M), *173*
Dupechecker (Apple), *176*
Dupechecker (Commodore), *180*
Dupesheet Printer (TRS-80 and CP/M), *187*
Dupesheet Printer (Apple), *190*
Dupesheet Printer (Commodore), *192*
General Purpose Contest Logger (TRS-80 and CP/M), *199*
General Purpose Contest Logger (Apple), *206*
General Purpose Contest Logger (Commodore), *214*
Field Day Logger (TRS-80 and CP/M), *237*
Field Day Logger (Apple), *243*
Sweepstakes Logger (TRS-80 and CP/M), *251*
Sweepstakes Logger (Apple), *257*
Log Printer (TRS-80 and CP/M), *266*
Log Printer (Apple), *267*
Log Printer (Commodore), *268*

ANTENNA PROGRAMS

Antenna Scaler (TRS-80 and CP/M), *276*
Antenna Scaler (Apple), *278*
Antenna Scaler (Commodore), *281*
Vertical Pattern Plotter (TRS-80 and CP/M), *287*
Vertical Pattern Plotter (Apple), *289*
Antenna Matching Evaluator (standard version*), *292*

TO THE MOON—AND BEYOND

EME System Calculator (TRS-80 and CP/M), *298*
EME System Calculator (Apple), *301*
EME System Calculator (Commodore), *303*
Moontracker (TRS-80 and CP/M), *310*
Moontracker (Apple), *312*
Moontracker (Commodore), *315*
Moontracker (Timex/Sinclair), *318*
Skylocator (TRS-80 and CP/M), *322*
Skylocator (Apple/Commodore*), *323*

CHAPTER

1

A GALLERY OF PROGRAMS FOR RADIO AMATEURS

To be included in this book, each program had to meet two requirements: (1) it had to do something worthwhile in the hamshack—and do it well—and (2) it had to run on several of the most popular brands of personal computers on the market today. This is not a collection of assembly language programs written for hardware that almost nobody owns. Nor is it an assortment of little programs that calculate Ohm's Law or resistor color codes. There's a place for that sort of thing, but it's easier to use a pocket calculator to do Ohm's Law problems, and most of us know the resistor color codes by memory—or at least we know where to look them up quickly. Goodness knows we'd never get anything done if we had to stop and boot up a computer each time we needed to know the value of a resistor.

Instead, this book contains BASIC programs that solve practical problems for radio amateurs: these programs do the kinds of jobs that computers are uniquely suited to do, jobs that involve processing lots of data or performing multiple mathematical calculations.

You'll need no specialized knowledge to get these programs going on the computers for which they were written. If you have a disk-drive-equipped Apple II, IBM PC, TRS-80, Commodore 64, or any computer that runs CP/M and Microsoft BASIC, you can use these programs as they are listed for your particular computer. Owners of the Commodore VIC-20 and Timex computers can also run a number of the programs.

The hardware requirements are straightforward. To make full use of the programs here, you will need 48K of memory, a disk drive, and a

printer. You can run the programs without a printer, of course, but many of them produce information that really should be in a hard-copy form: contest logs, dupesheets, charts showing beam headings from your station to countries all over the world, and the like. Without a disk drive and 48K of memory, the most complex programs cannot be run—but there are still a number of programs you can run even with the least expensive microcomputers.

As we planned this book, we realized that we would be addressing a diverse audience. Some readers are computer professionals with years of experience. But others are newcomers to both computers and amateur radio—people who are a little bewildered by the technical jargon so rampant in both fields. Therefore, we included programs that should interest both seasoned computer buffs and novices. And we have tried to provide documentation and explanations suitable for readers with these diverse backgrounds.

Because the early chapters are introductory in nature, we felt we should begin with an overview of the programs you'll find in this book. What follows is such an overview, highlighting the features of each.

Highlights of the Programs

The LOGBOOK—There are two general-purpose logkeeping programs in this book. One, presented as an introduction to BASIC syntax in Chapter 5, is something of a mini-logger. A one-page program, it is one that a good typist can have running in less than half an hour. It simply takes advantage of a computer's data handling capability by creating a data file to store records of radio contacts, with the ability to edit the entries, add new ones, and print out the results.

Of more interest to serious operators will be the full-featured program we call LOGBOOK, listed and described in Chapter 6. That logger also creates data files, retrieves them, and prints out the results—but in a much more sophisticated way. It will search through the data and separate out all QSOs (i.e., two-way contacts) with a particular station, on a particular band, on a certain date, with a certain prefix, or on a particular mode. It will print out only recent QSOs, or only those long ago. Did you ever wonder if you've worked someone before, perhaps some time back? This program will tell you that. And did you ever wonder who hasn't replied to your QSL? The LOGBOOK program will tell you that, too. In short, it brings the kind of data processing capabilities that businesses demand (and rely on) to ham radio log-keeping.

The RADIO AWARDS DATA BASE—This program also applies data-base-management techniques to amateur radio, but with an emphasis on the needs of those pursuing operating awards. If you're interested in five-band DX Century Club (DXCC), five-band Worked All States (WAS) or Worked All Zones (WAZ), or the new VHF-UHF Century Club (VUCC) Awards, this program was written for you. You can use it for any of these awards programs—or many others. It will handle hundreds of categories, be they

states, DXCC countries, counties, or grid blocks for VUCC, keeping track of which ones you have worked and confirmed on each band.

Suppose you hear a rare state or country on 20 meters. Are you sure you've got a QSL from that country on 10? How about 80? Do you have it confirmed on both voice and CW? Suppose you're a VHF DXer and you hear a distant and rare grid block on 2 meters. Do you need it on 432? Are you sure?

The AWARDS DATA BASE prints detailed reports showing where you stand in your quest for various awards. You can call up the results overall or band by band and state by state (or country by country, grid block by grid block, or however you choose). It will even alert you if you've been remiss about sending a QSL!

The GRID LOCATOR—The GRID LOCATOR is a program designed for the VHF-UHF Century Club awards program and the new VHF contest scoring system, both of which are based on a latitude-longitude grid system. A short and simple program that will run on the least expensive computers, it tells you what four-character "locator" any station is in. You enter the latitude and longitude; the correct locator will be computed and displayed.

The Worldwide "LATLONG" Data File—At the heart of four different programs, the LATLONG data file is a list of all DXCC countries and many major cities. It includes the prefix, country, state or city name, and the latitude and longitude. Once you have this data file safely stored on a floppy disk (or written into the programs themselves—we describe a

Computers are becoming an increasingly important part of a well-equipped amateur radio station. Here WB6IDK uses a computer to keep track of QSL information in her quest for the five-band Worked All States Award.

way to do that for those using cassettes rather than floppy disks), you can do many things with your personal computer. The next four programs represent examples of the possibilities.

The SUNRISE CHART—If you're interested in DXing on "top-band" (160 meters), 80 meters, or 40 meters, you know how important a role sunrise and sunset can play in radio propagation. With the SUNRISE CHART program and the LATLONG data file, you can print out a chart showing the sunrise and sunset times all over the world on any given day (during a big contest, for instance).

The GRAYLINE—The term *gray line* has become something of a household word among DXers. The gray line is the twilight zone that covers some part of the globe at any given time, favoring paths where both ends are within a few minutes of local sunrise or sunset. This program will tell you who is on your gray line. You enter the day and your own longitude and latitude. It computes your local sunrise and sunset times and then does the same for more than 400 other places around the world, comparing the times against local times. When the times coincide (you can specify anything from times that match perfectly to a 45-minute leeway on either side of local sunrise and sunset), the program prints out the name of the place and its sunrise or sunset time. You'll be amazed at some of the strange and exotic places having sunsets at the same time as your sunrise (and vice versa). Equally amazing is the way the pattern changes with the seasons.

The BEAMHEADING CHART—With this program, you can print out a customized beam heading and distance chart. It will tell you the distances and directions from your home to more than 400 places around the world, printing out the results or displaying them on your screen. Customized beam heading charts are not new, but they used to cost about as much as this entire book. With this program, you can prepare customized beam heading/distance charts for yourself and for everybody you know—gratis.

The DX DISPLAY—The DX DISPLAY is an on-line version of the custom beam heading and distance program, a program for which you don't need a printer. Some of the people who have tried it think it's one of the more impressive things a personal computer will do, although it isn't a very complicated program. It loads the latitude-longitude information for cities and countries around the world into memory. Then as you tune the DX bands, you can enter the standard prefix for any country you hear. Your computer will quickly search through its memory and tell you the name of the country as well as the distance and beam heading.

The DX CHECKER—This is a simplified version of the last two programs, designed to run on the least expensive personal computers. It performs the same mathematical calculations as the BEAMHEADING CHART or DX DISPLAY program, but without the large-scale data-handling provisions. You enter your own latitude and longitude and then

the latitude and longitude of any other point on the globe. The program computes the distance and direction from your location to any other.

The SUNRISE CALCULATOR—Another program that doesn't use any data files—and hence works well on a computer without disk drives—is the SUNRISE CALCULATOR. It displays or prints out the sunrise/sunset times for any point on the globe that has a sunrise (the north and south polar regions do not on many days of the year, of course). It will compute this data for as many consecutive days as you wish, producing a nice table of the results. This program is particularly useful for a DXer interested in working a specific path.

The DUPECHECKER—Of all the programs in this book, one of the most interesting for many radio amateurs is the DUPECHECKER, our contest duplicate-checking program. On most personal computers it will store more than 4000 call signs in memory, keeping track of them on a band-by-band basis.

What is most impressive is the speed with which its hashing algorithm searches the memory and determines if a call has been worked before during a contest. The DUPECHECKER will tell you within one or two seconds if a call is a dupe, even with 1500 or more calls in memory. We have seen other dupechecking programs, but never one that does the job so quickly and thoroughly in BASIC. Most programs that do this fast enough to be useful are written in assembly language, and thus can only be used with the particular hardware and operating system for which they were written.

Among its other features, the DUPECHECKER offers a quick multi-band lookup: it not only tells you if a call is a dupe, but also if you still need that station on any other band. In short, it does a multiband dupecheck in a moment's time. The DUPECHECKER will print a hard copy of all stations worked while the contest is under way, or that feature can be turned off. The program also saves the calls in data files so a dupesheet can be printed later.

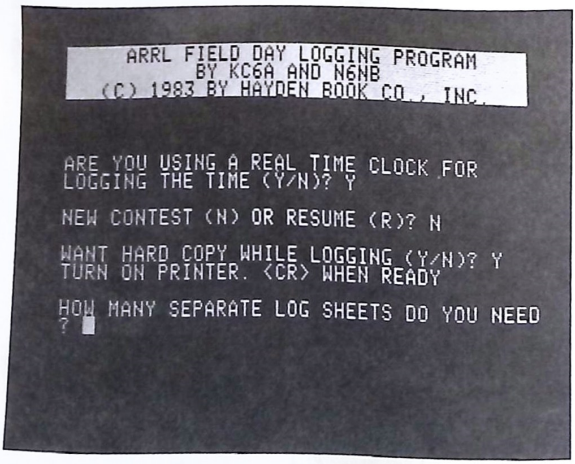
The DUPEPRINT Program—No dupechecking program would be complete without a companion program that sorts all of the calls and produces well-organized dupesheets. DUPEPRINT does a complete sort of the calls, either by call area or by prefix, alphabetizing the list. The resulting dupesheets are printed in a professional tabular format. The DUPEPRINT program also generates dupesheets for the contest logging programs.

The Contest Logger Family—This is a package of three comprehensive logging programs customized for major contests. Written by and for serious contest operators, these programs include all of the dupechecking features of the DUPECHECKER, but they go far beyond that. They provide instant running-score and multiplier lists on a band-by-band basis, along with provisions for full editing of any previous entry. Any and all portions of any contact can be changed, with the effect of

those changes on the multiplier and total score immediately calculated. If you thought you worked a station in a rare country or state and learn later that he's in a common one, you can edit in the correction and the logger will check to see if that was your only QSO in the particular state or country for multiplier purposes. When the contest is over, there isn't any of the post-contest drudgery you used to face. The log and dupesheet printing programs produce a complete scoring and multiplier summary sheet, along with the full log (without any dupes in it) and perfectly alphabetized dupesheets. You'll never again have to worry about being penalized for having uncorrected dupes in your logs.

The GENERAL PURPOSE CONTEST LOGGER—The GENERAL PURPOSE LOGGER is designed for VHF contests, DX contests, QSO parties, and almost any other contest where there are multipliers based on all or a portion of the exchange. If the score is determined on a QSO points-times-multipliers basis, the logger will give you an accurate running score at any time. You can choose separate multipliers on each band, or a format in which multipliers only count once regardless of the band. Likewise, each call can be dupechecked on a band-by-band basis or a once-per-contest basis. The logger will tell you pronto if you need a particular station on the band you're on—or any other! The point value of each contact can be varied from band to band, as is customary in VHF contests. Like the other loggers, this one will produce an on-line hard-copy log during the contest, along with a much neater version afterward, minus the changes, corrections, and revisions that inevitably occur during the contest.

The FIELD DAY LOGGER—The FIELD DAY LOGGER is a version of the GENERAL PURPOSE LOGGER that has been customized for the specific rules of the American Radio Relay League Field Day, traditionally


ARRL FIELD DAY LOGGING PROGRAM
BY KC6A AND N6NB
(C) 1983 BY HAYDEN BOOK CO., INC.
ARE YOU USING A REAL TIME CLOCK FOR LOGGING THE TIME (Y/N)? Y
NEW CONTEST (N) OR RESUME (R)? N
WANT HARD COPY WHILE LOGGING (Y/N)? Y
TURN ON PRINTER. <CR> WHEN READY
HOW MANY SEPARATE LOG SHEETS DO YOU NEED ?

This sign-on message is what you see first when you run the FIELD DAY LOGGER, one of three complete contest-logging packages included in Chapter 8. Field Day, held the fourth weekend in June, attracts about 25,000 participants—more than any other operating event on the amateur radio calendar.

the largest amateur radio contest in the world in terms of total participation. This program takes into account the fact that the multiplier is based not on sections or countries but rather on power levels and entry classifications, with CW QSOs worth more points than voice QSOs. It has the same band-by-band dupechecking features as the GENERAL PURPOSE CONTEST LOGGER.

The SWEEPSTAKES LOGGER—Few contests generate more fervor among their participants than ARRL Sweepstakes, and this logging program is designed for SS operators. It is tailored to the specific rules of SS, and even has a rapid "clean sweep" checker that allows you to pinpoint missing sections by looking them up in a table that lists all acceptable multipliers.

The LOGPRINT Program—A companion to the three logging programs, the log printer generates neat contest logs and summary sheets after the battle is over. Contacts may be presented in chronological order—as is preferred in most contests—or on a band-by-band basis as is common in certain specialty contests. After the contest, you run LOGPRINT and DUPEPRINT, and the paperwork of contesting is history.

The ANTENNA SCALER—No topic is of more universal concern among radio amateurs than antennas. The authors, who developed the first "Quagi" antenna, have a long-standing interest in the subject. We've seen many mathematical "formulas" for calculating antenna dimensions, but few of them can be counted on to work consistently in practice. There are enough random variables that computed dimensions are usually no more than a starting point for fine tuning. It's often better to find a design that has been field-tested and proven instead of computing the lengths from a formula.

If there isn't a suitable field-tested design available for a particular frequency, the next best bet is to find a design that has been field-tested and proven on another frequency and then scale it, correcting for changes in the length-to-diameter (L/D) ratio. Even this method is imperfect, of course, but it's the best we know of. The ANTENNA SCALER program does this kind of antenna dimension computing. You enter the dimensions of your favorite Yagi or cubical quad beam antenna, and it gives you dimensions for a new frequency, correcting for the L/D ratio if you wish. In addition, the program will save your favorite dimensions in a data file for later use, and it will print dimension charts after each run. But if you're willing to forgo these two features, the ANTENNA SCALER will run without disk drives or a printer.

ANTENNA MATCHING EVALUATOR—This short program will also run on almost any computer, telling you the hard facts about your antenna and transmission line. Simply enter a little readily available information, such as the SWR at the transmitter or the percentage of reflected power, and it will compute the losses in the system. The actual SWR up at the

antenna—and the small amount of your transmitter power that actually makes it there—may be a little surprising.

PHASED VERTICAL PATTERN PLOTTER—Written for any combination of phased vertical antennas, this program displays the radiation pattern. Enter the relative positions and method of feeding power to the antennas and the program will compute the pattern, not only on the horizon but at vertical angles up to 90 degrees too. Then it will take advantage of the high-resolution graphics of an Apple computer to plot the pattern on the screen. If you don't have an Apple, the program will still do a pattern plot on both the screen and your printer, using ASCII characters to show the shape of the radiation pattern.

EME SYSTEM ANALYZER—Earth-moon-earth (EME or "moonbounce") communication is one of the major challenges for radio amateurs today. It takes every bit of receiver sensitivity, transmitter power, antenna gain, and operator savvy you can muster to get a signal to the moon and back. This program tells you what you need to do the job. Enter all of your system parameters, and the program computes the signal level of the lunar echoes you can expect under typical conditions. Even if you're not into moonbounce, you may want to try various combinations of transmitter, antenna, and receiver performance just to see the results for different systems displayed (or printed if you wish). This is another program that will run without disk drives or a printer.

The MOONTRACKER—This new version of a classic computer program requires just the date and your location to compute not only moonrise and moonset times, but also tracking data as the moon makes its way from horizon to horizon. This program involves the most complex mathematical computations of any program in the book, and it produces a nice printout (or video display) of tracking data at regular time intervals. One entry will produce a printout for as many days as you wish; there's no need to separately type in a request for each day. Although mathematically complex, the program will run on inexpensive computers; we show how to make a simplified version of it run on a Timex/Sinclair 1000 (ZX-81).

The SKYLOCATOR—Many radio amateurs these days look not only at the moon but also far beyond, aiming dish antennas with sensitive preamplifiers at remote celestial objects. This program computes tracking data for any celestial body outside our solar system, based on its right ascension and declination.

These are the programs you'll find in this book. All are listed with annotations for various kinds of computers. But before we present the programs, we will be devoting several chapters to an overview of personal computer hardware and software, written primarily for newcomers.

CHAPTER

2

THE EVOLUTION OF THE MICROCOMPUTER

If you're a radio amateur, you already know a lot of things that most people have to learn the hard way when they adopt personal computing as a hobby.

To begin with, you know something about electronics—you've learned some theory, and you have some practical experience applying that theory in your hamshack. For instance, you probably know about interfacing equipment. You also know the difference between a transistor and a "chip," and you know which end of a soldering iron gets hot. The chances are you know far more than that, but even if that's all the electronics you know, you have a big advantage over the computer hobbyist who doesn't know a resistor from a capacitor, and who thinks Ohm's Law is an act of Congress named after a Senator Ohm.

If you've been a radio amateur for very long, you probably also know a little of the history of computers. Both ham radio and computers have been greatly influenced by the development of transistor and integrated circuit technology in the last 30 years. The steady reduction in the size and cost of electronic components—and the simultaneous increase in their reliability—have led to major improvements in both computers and radio gear.

You can be an avid radio amateur without necessarily knowing the difference between a PROM and an EPROM, or knowing why so many personal computers use "floppy disks" instead of "hard disks." If you do know about such things, you're even further along the road to success in personal computing—but you'll have to bear with us while we explain the history and basics of computers in the next two chapters.

The Birth of the Computer

Not too many years ago, every computer was a monstrous machine that lived in a temperature-controlled room—or building. If someone sneezed near one, it was almost like a national emergency. The computer was a mysterious beast that spoke strange languages—languages understood only by the white-coated wizards who were its caretakers. Many of us kept our distance from those strange people and frightening machines, content to work DX on 20 meters and satisfy our technical inclinations by building amplifiers or antennas.

But computers have changed their image lately, and changed it a lot. In the years since the first vacuum-tube "electronic brains" were developed in the 1940s, computers have gone through generation after generation of evolution, becoming ever smaller and more powerful—but also friendlier to us mortals. Your pocket calculator is in some respects more powerful and faster than the first electronic computer, a 30-ton monstrosity with 18,000 vacuum tubes.

Even so, that first computer—named ENIAC (Electronic Numerical Integrator and Computer)—was a miraculous machine. And it was even something of a miracle that it was built. When ENIAC was first proposed to the U.S. Army during World War II, a clever mathematician explained why it couldn't possibly work. Since the average life of a vacuum tube was 3,000 hours, and the beast would have 18,000 of them, a tube would fail at least every 15 minutes, he said. Since it would take more than 15 minutes to locate one bad tube among 17,999 good ones, the machine would never run long enough to do any useful work. But ENIAC did work, performing

In a striking contrast of technologies, a 3-500Z transmitting tube—a mainstay in kilowatt amplifiers for many years—stands beside a very small computer: the Sinclair ZX-81 (Timex 1000). The entire computer is barely bigger than the tube, and the tube costs more than the computer. But not that long ago, a monstrous analog computer with thousands of vacuum tubes couldn't do as much as the computers that are sold at drugstores today.

mathematical calculations about 1,000 times faster than ever before. It went into service in 1946 and was shut down for the last time in 1955.

That success was not without its price, however. For one thing, the initial development of ENIAC cost the government about $500,000 at a time when a "penny postcard" really cost a penny. Further, the 18,000 tubes and associated circuitry consumed 200,000 watts of electric power. That would have generated a fairly respectable signal on 20 meters (until the FCC came along), had the ENIAC folks chosen to put all that power into some very large transmitting tubes instead of row after row of receiving-type tubes.

After ENIAC's initial success, computer technology exploded. Two of ENIAC's designers, John W. Mauchly and J. Presper Eckert, Jr., formed a computer company that was quickly merged into Remington Rand. That alliance led to the UNIVAC I (for *UNIV*ersal *A*utomatic *C*omputer), which captured the public's attention by correctly predicting the outcome of the 1952 presidential election. UNIVAC I was a significant improvement over ENIAC, but it was still a giant machine with thousands of vacuum tubes. Nevertheless, it proved the practical usefulness of the electronic computer, at least in applications where cost didn't matter.

IBM Enters the Picture

Meanwhile, Thomas J. Watson, Jr., the young president of International Business Machines Corp., persuaded his reluctant father—by then the chairman of the board at IBM—to let him take IBM into the computer business. The elder Watson felt that computers had few commercial possibilities due to their outrageous cost. Perhaps eight or ten computers would saturate the entire market, he predicted. But his son prevailed, and in 1953 IBM introduced the model 650, which quickly became the industry leader with 1,000 units in place—100 times the sales volume Watson senior envisioned.

However, we're still talking about bulky, expensive, unreliable vacuum tube computers. IBM introduced its first all-transistor computers in 1958. These "second generation" machines were faster, smaller, cheaper and more reliable than the vacuum tube units they rendered obsolete. By 1964, there were nearly 20,000 "second generation" computers in service, performing an incredible variety of mathematical, scientific, and business-oriented tasks for government agencies and private industry.

Nevertheless, these machines occupied large rooms, required special temperature controls, and were very difficult to operate. Although two of today's most popular programming "languages" (FORTRAN and COBOL) had been developed by then, computer programming remained a nightmare. First, a trained specialist had to write a program. Then instruction cards had to be punched (one instruction per card), a task which might take several days to accomplish. Next, the cards were fed into the computer to see if the program would run. If it didn't (and programs almost never did on the first try), the programmer had to figure out what was wrong, have more cards punched, and then repeat the whole process.

If you've ever written a program in BASIC on a modern personal computer—a computer that allows you to type in the instructions, run the program, and find out immediately if there's a "bug"—you can appreciate just how convoluted this process was. Modern personal computers offer "interactive" programming: the computer listens to what you tell it and immediately responds. We'll say more about programming methods later. For now, suffice it to say programming today is immeasurably easier than it used to be.

But these early computers worked. And by 1964, most large companies and major universities owned or leased computers and had data processing staffs. Meanwhile, IBM's competitors began to cut into its market share with transistorized computers that were every bit as good as the early IBM systems.

The IC and the "Third Generation"

Despite the emergence of new competitors, anyone who thought IBM was losing its grip on the computer market in 1964 was badly mistaken. In April of 1964, IBM again turned the computer world upside down, announcing the revolutionary System/360 family of computers. These computers used an astonishing new breakthrough in electronic technology—the integrated circuit—in place of the discrete components used in the earlier solid-state computers. The result of this "third generation" of computer technology was another dramatic reduction in size and cost for a given level of computing power—and another increase in speed and reliability. For example, "first generation" vacuum tube computers could typically perform 2,500 computations per second at a cost of about $1.40 per 100,000 computations. The largest System/360s could perform 375,000 computations per second at a cost of roughly four cents per 100,000 computations.

Dramatic as it was, this new development didn't occur overnight. Industry analysts estimated that IBM invested $5 billion—about one twentieth of what Uncle Sam spent to run the entire federal government in 1964—to develop the System/360. Fortunately for IBM, the gamble paid off: the company delivered 33,000 System/360s by 1970.

Nevertheless, IBM faced increasingly stiff competition from upstart rivals in the computer business. For instance, the Digital Equipment Company (DEC) announced its first "minicomputer," the PDP-8, in 1965. That computer took full advantage of ICs to deliver large-mainframe performance in a much smaller package. Because of the PDP-8's modest size (and relatively low price), these minicomputers began showing up in laboratories, schools, and medium-sized companies that could never have afforded their own large mainframe computers.

But even that was only the beginning. The 1960s saw a massive federally funded research and development program to put mankind on the moon. The project was a spectacular success, not only because it led to the historic "giant step for mankind" of the 1969 moonwalk, but also because of its fringe benefits for electronic technology. By 1970, the

density of solid-state "chips" had been increased enormously. The term *large-scale integration* (LSI) came to refer to any chip with 1000 or more logic gates, while *very-large-scale integration* refers to a chip containing 10,000 or more gates. At last, the modern microcomputer was becoming feasible.

The Microprocessor Chip Arrives

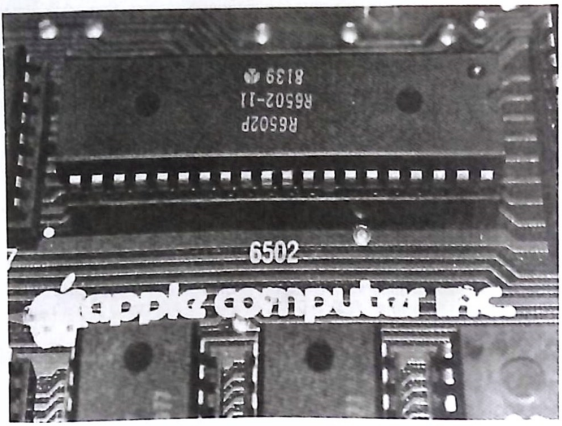
In 1970, the Intel Corporation began marketing its 8008 chip, which it enthusiastically called "a computer on a chip." The 8008 was still primitive by present-day standards, but it ushered in the modern era of single-chip central processing units (CPUs). Within a few more years, Intel had developed the 8080 CPU chip, Motorola offered the 6800, Mostek announced the 6502, and Zilog produced the Z80 family, each upstaging the others in some respects. With rival manufacturers rushing to produce ever more sophisticated, densely packaged, and inexpensive CPU chips, the result was a bonanza for design engineers—and end users. Any of these chips can be purchased for less than ten dollars today, and each offers far more central processing capability than the entire 18,000 tubes of ENIAC.

Another remarkable comparison between ENIAC and a modern CPU chip is the power consumption. To put it in amateur radio terms, ENIAC was a 200-kilowatt rig, but a 6502 or Z80 runs on about one watt, which definitely puts it in the QRP class.

Computer memory components have also undergone a dramatic reduction in size and cost. The early mainframes used "core memory"—memory components consisting of toroid-like ferrite beads with wires passing through them. They were replaced by memory ICs in the late 1960s. By 1980, Japanese manufacturers were producing memory chips with more than 65,000 individual electronic switches, all crammed on a "wafer" smaller than your little finger. A memory bank of 256,000 bytes (a byte is eight individual bits of information) could be built on a single printed circuit board small enough to fit inside a desk-top computer.

In the mid-1970s, the revolution in chip technology had made it practical to build a full-featured computer on a hobbyist's budget. Enterprising hobbyists all over the world (many of them radio amateurs) began building powerful home computers. In 1975, *Popular Electronics* magazine carried a construction article on the ALTAIR 8800 personal computer system.

Not only was the ALTAIR an excellent computer, but it also pioneered a concept that has been widely adopted in more recent designs: plug-in PC boards for the CPU, memory, and other functions. The ALTAIR used a "motherboard" with 100-pin edge connectors which accepted the individual PC boards. As other manufacturers began using the same arrangement and the same pin configuration, this approach came to be called the "S-100 bus." It is one of the most popular types of personal computer construction today, another being "single-board" design. In a single-board computer, everything—CPU, memory, input/output circuitry,



This is an ant's-eye view of the main circuit board in an Apple II computer, showing the heart of the machine—the 6502 microprocessor chip. When it became possible to put a whole "computer" on a chip, a revolution in technology began.

and sometimes even disk controller logic and a power supply—is mounted on one circuit board.

While some hams were building ALTAIRs, another radio amateur (Wayne Green, the publisher of *73 Magazine*) was creating the first specialty magazine for computer hobbyists, *Byte*. The rush into this new technical hobby had begun—with many radio amateurs in the forefront.

The Microcomputer Boom Begins

Within the next few years, a flood of commercial microcomputers appeared on the scene as entrepreneurs hurried to capitalize on the new technology. The Tandy Corporation, owner of Radio Shack and a longtime leader in consumer-oriented electronics, announced the original TRS-80 personal computer, the Model I. Almost simultaneously, Commodore Business Machines, Inc., shocked the computer world by offering the PET computer for what then seemed like a give-away price, $495. However, that price was quickly increased, and would-be purchasers faced long delays in delivery. Radio Shack, on the other hand, offered much faster deliveries and marketed the TRS-80 through its vast network of neighborhood electronic stores. As a result, the TRS-80 Model I quickly gained a dominant position in the personal computer market.

The Heath Company, an old-line manufacturer of consumer electronics and amateur radio gear in kit form, introduced a line of personal computers. The H-89 "all-in-one" computer, which originally sold for $1695 in kit form (including both a disk drive and an industry-standard 24-line, 80-column video display) was especially popular. In fact, Zenith entered the personal computer field by marketing the factory-assembled version of the Heathkit H-89, calling it the Z-89.

Before long, the TRS-80 had many other large and small competitors in the personal computer marketplace. For instance, the success story of

Apple Computer, Inc. is a legendary example of the American dream in action.

Apple's Emergence as an Industry Leader

Working in a garage in 1976, two college dropouts barely 20 years old built the first Apple computers in California's Silicon Valley. The Apple II caught on in an unprecedented way. By 1980, their company had $200 million in sales and was producing at least 150,000 computers a year. In four years, a tiny company had come from nowhere to equal and then surpass the giant Tandy Corporation's market share of the small computer business. By 1982, Apple had topped half a billion dollars in annual sales, and *Time* magazine was calling the Apple II the "Model T of the computer industry."

Perhaps the secret of Apple's success was the same as Radio Shack's: Apple offered the basic computer at a modest price to get the hobbyist started, and then sold all of the niceties (such as disk drives for data and program storage) as extra-cost accessories. And like Radio Shack, Apple offered a video display that was strong on graphics but contained fewer lines and columns of characters than the industry standard. This obviously appealed to hobbyists, but it frustrated small business and word processing users. As supplied by the factory, the original Apple II could display the standard number of lines (24), but only 40 columns, not the standard 80. The TRS-80 Model I offered a 16-line, 64-column display.

Larger CP/M Systems

In the late 1970s, many other entrepreneurs plunged into the marketplace. At the expensive end of the personal computer market, Cromemco, Northstar, Vector Graphic, and others were offering desk-top "mainframe" computers, complete with built-in disk drives, 64 kilobytes of random access memory, and sophisticated input-output device capability. Unlike the less expensive Apple and TRS-80 systems, most of these computers worked with any professional video display terminal (VDT). The same VDTs that were used with giant time-sharing computer systems would work with these microcomputers, something that must have been an important psychological factor for hobbyists with big-computer experience.

Also, these larger microcomputers could run a "disk operating system" that became the de facto industry standard, Digital Research's *CP/M* (for Control Program/Microcomputer). A great deal of software was written for CP/M-compatible systems, and anyone who knew how to operate one CP/M computer could run any of the others without having to learn many new commands.

Attesting to the popularity of the 24 × 80 video display and the CP/M operating system, thousands of Apple owners have purchased plug-in

accessory boards to make their computers display 80 columns and run CP/M—even though the add-on CP/M board supplants the heart of an Apple computer (the original 6502 CPU chip) and uses a Z80A CPU. The result, of course, is that the modified Apple isn't exactly an Apple any more, but it will run CP/M-based software as well as Apple software. This gives the user access to the two largest banks of off-the-shelf software available anywhere at this writing.

Meanwhile, Radio Shack added the TRS-80 Model II, the Model III, the Model IV, the Model 12, the Model 16 and other new computers to its line. The Model II, Model 12, and Model 16 were targeted a little above the hobbyist market, however. The Model 16 was introduced with a $5000 price tag, for example. But for that price, the customer got a computer with both a Z80A and a 16-bit MC68000 CPU. This preserved software compatibility with 8-bit systems while also offering the greater speed and memory capacity of a 16-bit CPU. These bigger TRS-80s offered many business-oriented features, such as a standard-size video display and 8" disk drives with much more storage capacity than was typical of 5¼" drives. Most were capable of running standard CP/M software without extensive modifications, too.

The Bargain-Basement Computer

At the other end of the microcomputer price spectrum, England's Ian Sinclair, a consumer electronics marketing maestro, introduced the Sinclair ZX-80 and quickly followed it with the ZX-81. Within about a year, some 250,000 of these little computers were sold, and before long the ZX-81 was selling in kit form for $99.95, including an "RF modulator" so an ordinary television set could be used as a video monitor. The ZX-81 came complete with a version of BASIC in ROM (read-only memory). Sinclair even threw in the interconnecting cables for a cassette recorder so programs could be stored on tape.

Then in 1982 Timex—the famous low-cost watch company—took over marketing of the ZX-81, renamed it the Timex/Sinclair 1000, and dropped the assembled price below $100. For another $50, the customer could add a 16K plug-in memory module. Before long, the computer itself was selling for $50 or less. Suddenly, computers were being sold alongside Instamatic cameras and clock radios in America's drug stores! And judging by the conversations on the amateur bands, two thirds of the hams in America owned at least one computer by this time.

Not to be outdone, Atari quickly moved up from the video-game market—in which it had a dominant position—to offer personal computers. The Atari 400 was designed to compete in the low price range: its price was cut repeatedly until it sold for only about twice the price of the Timex/Sinclair, even though it came complete with 16K of RAM. The BASIC language, however, was an extra-cost accessory. Atari also marketed computers designed to compete with the Apple II, including the model 800 and a restyled model 1200. They offered many of the same features as the Apple, including a 6502 CPU chip and a 24 × 40 display,

but their retail price was somewhat lower, and these computers lacked the expandability and software selection that had made Apple computers so attractive. In the early 1980s, Atari's share of the personal computer market was growing, but perhaps not as fast as the company's stockholders had hoped. Atari's video-game image apparently did not enhance the marketability of its more powerful computers.

Commodore rejoined the hobby-computer sweepstakes in 1982 with a big push for its VIC-20 and Commodore 64 computers, which were heralded as two of the best buys on the market. By 1983, the VIC-20 generally sold for less than the Atari 400, even though it had BASIC built in and its keyboard had "real" individual keys, features the Timex/Sinclair 1000 and Atari 400 both lacked. And the Commodore 64 offered a full 64K of RAM, good color graphics, and versatile sound capabilities in a package initially priced below $600. It wasn't long before some dealers were selling the Commodore 64 for one third of that. With Commodore's low-cost add-on disk drive and printer, a complete disk-oriented system could be up and running for a fraction of what such as system would have cost only a few years earlier. Advertisements for the VIC-20 began showing up regularly in ham magazines, and several manufacturers marketed RTTY-CW interfacing hardware and software for it.

Still another price breakthrough occurred in late 1981, when Adam Osborne, long known for his computer books and accounting software, announced the Osborne I with a list price of $1795. What was astonishing is that the price included a compact CP/M-based computer with two built-in disk drives, 64K of memory, and a very small but self-contained video monitor—plus specialty software that would normally sell for nearly as much as the computer itself. By comparison, few other manufacturers were including anything except their own proprietary operating systems and perhaps BASIC in the base retail price. In 1983, Adam Osborne shocked the computer world again by filing for bankruptcy, and other portables replaced the pioneering Osborne I.

Amidst all of these new-product announcements, there were recurring rumors that Apple was planning to launch a new line of personal computers. The Apple III had been announced in late 1980, but the early models had problems. Although these problems were corrected, the Apple III never quite captured the public's imagination the way the Apple II did. Apple fans waited through a series of "inside" stories about what new magic might be in the works at Cupertino, but nothing new was announced—for a time. But at the Apple stockholders' meeting in January 1983, co-founder Steven Jobs showed up in blue jeans to address the assembly of *nouveau riche* millionaires—and unveiled two new Apple computers.

The Apple IIe ("e" for "enhanced") replaced the venerable Apple II, and a much more expensive computer named "Lisa" was targeted for the business computer market. The IIe featured a much-improved keyboard, more built-in memory, a low-cost 80-column option, and a redesigned main board, among other features. But in many ways, it was still the familiar old Apple. In fact, Apple went to great lengths to keep the IIe compatible with the 20,000-odd programs that had been written for the

Apple II, a feature that was hailed by computer hobbyists. But Lisa (and later, the Macintosh) was unapologetically aimed at users with little or no interest in the inner workings of a computer—its main selling point was its appeal to non-computer types.

The Giants Enter the Field

If little guys were making fortunes in the personal computer business in the late 1970s and early 1980s, the barons of the large-computer business were not unaware of what was happening in small computers, either. In 1981, IBM announced its long-awaited (and, by some, long-feared) entry into the personal computer market with a system designed to compete at once with the low-cost systems such as Apple and with the more expensive small mainframes. The IBM PC was greeted with considerable enthusiasm by small-business users. However, when other manufacturers looked at the IBM's specifications, they breathed a collective sigh of relief: IBM's system indeed had nice features, but it was by no means the revolutionary breakthrough some had expected from THE computer company. And the IBM PC wasn't intended to compete with the Sinclair 1000 or VIC-20, either: the typical model sold for at least $3000. Nevertheless, by early 1983 IBM had captured about 15 percent of the microcomputer market, a share that rapidly grew as the PC became the de facto industry standard—forcing others to adopt the MS-DOS operating system, a sophisticated descendant of the CP/M operating system. In 1984, IBM began selling the PC*jr* in the home computer market.

Meanwhile, Xerox—a company that had backed out of the large computer business a decade earlier—entered the small computer field with the Xerox 820, a system that used the CP/M operating system as its standard and was positioned to compete at the high end of the personal computer market. DEC joined the battle with its "Rainbow" line of small business computers, offering both high performance and attractive packaging. And dozens of other manufacturers plunged into the marketplace to compete for the consumer's dollar at the same time, prompting *Business Week* to predict a massive shakeout in the industry within the next few years.

While all of this was happening, several of Japan's electronic giants were also looking at the American personal computer market with more than a little envy. Among the Japanese pioneers were NEC (the Nippon Electric Company) and Toshiba, both of which announced attractive personal computer lines in 1981. However, both had a problem: almost everyone who has ever tried to market personal computers says it's the software and not the hardware that sells computers to the public, and the Japanese systems lacked sophisticated applications software tailored for the American market. As a solution, NEC made its system CP/M-compatible, seeking to tap the vast storehouse of CP/M software that was already available. On the other hand, Epson—the Japan-based firm that had captured a giant share of the low-cost printer market—announced a do-everything personal computer that was designed to be at once very

powerful and very user-friendly with a take-the-user-by-the-hand operating system. How any of these products will ultimately do in the American marketplace remains to be seen.

The leading American electronic calculator manufacturers were also moving into the small computer business in the early 1980s. Texas Instruments launched and then dropped the low-priced 99/4, and TI and Hewlett Packard both challenged the IBM PC with higher-priced personal computers that used the MS-DOS operating system.

Boom Times—In Computing and Radio

With all of this action, the personal computer business was enjoying unprecedented public attention in the early 1980s. Indicative of the boomtown mood, *Time* magazine put 26-year-old Steven Jobs of Apple Computers on its cover—a place usually reserved for heads of state and celebrities. And then instead of announcing a "Man of the Year" as it usually does, *Time* named the personal computer the "Machine of the Year" in 1982. All of the major media were covering trends in small computers as big news, and the major computer manufacturers were advertising their products on prime-time television. Personal computing had captured the public's imagination in the early 1980s much as radio did during its golden age 50 years ago.

Thanks to its colorful and innovative young entrepreneurs, personal computing today is very much like amateur radio in the 1920s and 1930s. Old-timers who remember the pre-war radio entrepreneurs—bright and talented young people like W.J. Halligan (of Hallicrafters), Art Collins (of Collins Radio), and Bill Eitel (of Eimac)—can see many parallels between their exploits and those of the daring and technically skilled young pioneers of today's personal computer industry.

This inexpensive and compact Commodore 64 system takes up only a little space at the operating position, but it will do many things to make hamming more enjoyable and efficient.

Like amateur radio in the early 1930s, personal computing today is a technical hobby whose time has come, and whose devotees are discovering exciting career opportunities as an outgrowth of their hobby—at a time when much of the economy is struggling through hard times.

Many of us can only read with envy of the exploits of radio amateurs in the early days. What a thrill it must have been to work coast to coast on 20 meters in 1924, to work all continents in the 1930s with a two-tube transmitter, or to hear astonishing DX rolling in on the "line of sight" five-meter band—and to work some of it with a modulated oscillator! Imagine what it must have been like to hear a ham operator in England or Japan on a regenerative receiver, and then to frantically call him, even though your crystal was 100 kilocycles (NOT kilo*Hertz*) away from his frequency—and to hope your chirpy signal didn't drift away before you signed your call.

In this world of store-bought SSB kilowatts, pre-drilled and pre-marked antenna kits, frequency synthesizers, and repeaters, it's hard not to wonder if some of the early spirit of amateur radio has gotten away from us. Modern radio communication is so dependable—and the hardware leaves the operator with so little to do but *talk*—that some find it all a little boring. But in the merger of microcomputers and amateur radio, thousands of hams have rediscovered some of the pioneering spirit of a bygone era. A lot of us have learned that it's still exciting to be near the state of the art—at the cutting edge of a new technology on the move.

If you're someone who knows that thrill, we hope you enjoy using and customizing the radio-oriented computer programs in this book as much as we enjoyed writing, compiling, and debugging them.

CHAPTER

3

MICROCOMPUTER HARDWARE

"And the rig here is a Drake TR-7 driving an L-7 amplifier to the . . . legal limit. The antenna is a Hy-Gain TH6DXX on a 70-foot crankup tower. . . ."

Sound familiar? To most radio amateurs, it would. But if you described your station in those terms to a non-ham friend, you'd probably get a blank stare. To him, it would be undecipherable technical jargon—shop talk.

Here's another equipment description:

"I'm running CP/M on an Apple II with a Z80 card, a 16K RAM card, an 80-column card, a Sanyo monitor, and two 5¼-inch drives. My printer is an Epson MX80 and I have an Apple-Cat modem."

Does that one sound familiar?

Like a ham radio station, a personal computer system consists of a lot of hardware, all interconnected with cables. And like amateur radio, personal computing has its own technical jargon, a vocabulary that bewilders outsiders. If you're an experienced "hacker" (a term for a computer hobbyist that seems to be catching on much as "ham" did with radio amateurs), you already know all of this. However, if you're a computer newcomer, perhaps this chapter will make the jargon of computer hardware a little less mystifying.

The CPU and the Desk-Top Computer

If a transceiver is the heart of a ham radio station, the central processing unit (CPU) is the heart of a computer. Stripped to their essentials, most personal computers these days consist of a CPU on a single chip, some input/output circuitry, both random access memory (RAM) and read-only memory (ROM), and a keyboard, all in a neat desk-top package. Most personal computers are sold in this fashion to keep the price low, allowing the user to add accessories as his or her budget and interest dictate.

Probably the two most popular CPU chips are still the 6502, used in the Apple II and Atari computers (among others), and the Z80, used in the Radio Shack TRS-80, Timex/Sinclair 1000, Heath H-89, and many of the larger "mainframe" personal computers. Of these two CPU chips, the Z80 has a slightly larger instruction set and is a little faster, but both offer good enough performance that most computer users would be hard-pressed to tell the difference between them. Computer purists can tell the difference, of course, and they'll also point out that there are other microprocessor chips that are faster and have better instruction sets than either of these two. And the "16-bit" CPUs are much more powerful than any of these "8-bit" CPU chips. Still, in many personal computing applications an "8-bit" chip like the Z80 (or 6502) is quite satisfactory.

An 8-bit CPU is so designated because its *data bus* is eight bits wide. That is, there are eight data lines, each of which accommodates one bit of each byte the CPU sends or receives. The data bus is the main highway through the computer, carrying information from the CPU to the memory, disk drives, and external peripherals such as a video monitor or printer.

A 16-bit CPU has 16 data lines rather than just eight. As a result, it can transfer a given amount of data about twice as fast. However, a 16-bit CPU is a larger chip, and the double-width data bus complicates the hardware requirements and increases the cost.

Thus, manufacturers will probably continue to build many personal computers around 8-bit CPUs for some time to come, despite the greater speed and memory-handling capacity of 16-bit microprocessors. Other factors weigh more heavily than the specifications of the CPU chip when most consumers select one personal computer over the others.

Among these other factors, some are subjective and others may depend on a person's specific application. But several things are important to almost every personal computer hobbyist: expandability, the video display and keyboard, software availability, and price. A vast amount of software has been written for the Apple and TRS-80 systems by now. Likewise, much has been written for the systems that run CP/M. If anything, there is more business-oriented software for CP/M, with personal and game-oriented software more plentiful for the Apple.

By 1983, the IBM Personal Computer (PC) had also become a major force in the microcomputer industry. IBM's market share was growing rapidly, and much software was being written for the PC. Meanwhile, almost as many manufacturers were making plug-in accessory boards for the PC as for the Apple II. But perhaps even more impressive, a number of

other manufacturers were marketing IBM-compatible computers—computers that would run IBM software and read IBM PC-format disks without modification. Built around the 8088 microprocessor, the PC offers users many advantages of a full 16-bit CPU, but without the complexity and cost (the 8088 is a 16-bit microprocessor internally, but it has only eight data lines).

Expandability and Other Factors

Aside from the availability of software, expandability may be the most crucial thing to consider in choosing one personal computer over another. Although you may not expect to upgrade your system, there's nothing more frustrating than learning that you're locked into a system that won't grow to keep pace with your growing interests and needs. Any system that doesn't allow you to add at least two disk drives, 48K of RAM or more, and a variety of printers or modems may become a white elephant to a serious hobbyist. A nonexpandable computer is a little like a two-channel crystal-controlled radio transceiver: it's fine until you get bored and want to do new things with it.

This is not to say there isn't a place for a computer with only limited expansion capabilities—provided the cost is low enough and you understand the limitations before you buy. The Timex/Sinclair 1000, for instance, will do an amazing number of things for its low price. But it won't do many things the more expensive systems will do: it has no provision for disk drives, for example. And with its small, nontactile keyboard, it would hardly be suitable for any application that involves extensive input typing. Also, its version of BASIC lacks not only provisions for data files, but also the READ and DATA commands. In addition, its string-manipulation capabilities are very limited. However, the Sinclair may be a good first computer for a young person or second computer for a more experienced hobbyist. But for a little more money, you can get a computer such as the VIC-20 that has many of the features the Timex/Sinclair lacks. It all depends on what you want your computer to do.

As we mentioned in Chapter 2, a major consideration for some computer buyers is the video display capability. Most of the inexpensive personal computers require an external video monitor (or a TV receiver and an "RF modulator"). Some systems offer color graphics, while others do not. Few personal computers offer the industry-standard 24-line, 80-column display. The standard Apple II and Atari 400/800 display is 24 lines, but only 40 columns, while the TRS-80 Models I and III have a 16-line, 64-column display. However, Apple offers an 80-column board for the IIe, and several independent manufacturers have marketed competing 80-column boards for the Apple II family.

The Sinclair 1000 and Texas Instruments 99/4 both use a 24-line, 32-column display. The Commodore 64 features a 40-column display, but the VIC-20 has perhaps the smallest display window in the industry: 22 columns.

The Heathkit H-89 was among the early few modestly priced microcomputers that offers a full-size 24 × 80 display as standard equipment. Most of the small mainframe systems use external terminals that provide a 24 × 80 display, but the terminal must be purchased separately. And few of the systems offering a 24 × 80 display can match the graphics capabilities of the smaller personal computers. This means you may have to trade off a full-size display for color graphics.

The programs in this book will work correctly on a 40- or 64-column video display. Many will work with even narrower text windows. The 80-column video format makes the charts and tables produced by some of the programs more attractive and more complete, but most users will probably want a full-size hard copy of these tables anyway. A list of beam headings and distances, sunrise/sunset times, or moon tracking data is something most users would want to keep and consult periodically, not just see once on a video display. And by definition, a contest log or dupesheet is still a hard copy: few if any contest-judging organizations are geared up to accept logs in the form of data files on floppy disks submitted by participants. For these reasons, an 80-column display isn't vital for most personal computer users; trading off a professional-size display for enhanced color graphics (and bigger type on the video screen) may not be a bad deal.

Another trade-off is the choice of a single-board computer as opposed to the "S-100 bus." The S-100 bus, with its "motherboard" arrangement and 100-pin edge connectors, allows the user to add a wide variety of boards to the basic system. S-100 bus systems (such as the Northstar Horizon, Vector Graphic MZ and Cromemco System Three) tend to be more expensive than single-board designs, and they may be a little less reliable in the long run due to the possibility of dirty connector contacts. In the early days of personal computing, S-100 bus systems tended to be the standard, but their higher cost has led a growing number of home computer users to opt for single board designs. Nevertheless, for a serious computer hobbyist, the S-100 bus still offers more flexibility and expandability than most single-board systems.

Even among single-board systems, expandability varies considerably. The original Xerox 820, for instance, had a single board with no provisions for additional plug-in boards. On the other hand, one of the best features of the Apple II is its expandability: it has eight slots so the user can add various plug-in boards, including many that are made by other manufacturers. The Atari and TI 99/4 computers take still another approach: the first accessory is plugged into the computer, the second is plugged into the first, and so on.

The outboard plug-in arrangement is convenient—up to a point. But what happens if someone bumps an add-on accessory while the system is running? Obviously, there is a potential for intermittent connections in a long chain. In fact, it's possible to have disasters with even one outboard accessory plugged directly into a personal computer by means of an edge connector, as we've learned with the 16K memory module plugged into a Sinclair 1000. If the module is bumped, the program is likely to crash even if the module doesn't come unplugged.

Even typing on the Sinclair sometimes causes momentary breaks in the connections to the memory module, with resultant program crashes. You learn to type very gingerly with such a system. One solution is to secure all of the units to the table top or a board so none can move in relation to the others—and type with a light touch. Obviously, there is something to be said for having the accessories safely mounted on expansion slots inside the computer, not plugged in as appendages on the sides and back of the unit.

For some personal computer users, the keyboard is another important consideration. People who are accustomed to the "secretarial" keyboard layout standardized on the IBM Selectric typewriter line (and used on most professional terminals) tend to find the keyboards on small computers annoying. Not only may the keyboard have a strange "feel," but a number of the keys aren't "where they're supposed to be." We know one college professor (a fast touch typist) who bought an Apple II-plus mainly for word processing and discovered belatedly that some of his favorite keys weren't where he expected to find them. That made him angry, but when the Apple IIe came out a few months later—complete with the standard keyboard layout that he had wanted all along—he was ready to lynch someone in Cupertino.

Some microcomputers also have another problem: keyboard bounce. You may think you only struck a key once, but the character shows up twice on the screen. Early TRS-80s often had this problem, but it's been corrected in the more recent TRS-80 models, which feature a keyboard that fast typists tend to like for its feel if not for the arrangement of the keys. At the other extreme from the keyboard-bounce problem, some keyboards strenuously resist the typist who wants to type the same letter twice in a row.

A suggestion: if you're shopping, make sure you do some serious typing on the computer you're considering. Since you talk to the computer by typing, being comfortable at the keyboard is vital.

RAMs, ROMs, PROMs and EPROMs

It's been said more than once that in the good ole' days the Rams were a football team and the Prom was a big dance. But in the computer business, acronyms like RAM, ROM, and PROM have unique meanings.

Virtually every small computer uses both random access memory (RAM) and read-only memory (ROM) for data storage. Actually, the term "random access memory" is misleading, because ROM may also be accessed more or less at random.

The crucial difference between the two is that RAM is *volatile:* that is, if you turn off the power, it goes away. Also, it can be changed at any time on command from the CPU. ROM, on the other hand, is nonvolatile memory: it cannot ordinarily be erased. Read-only memory is primarily used for a computer's basic operating system or for a program that is always present when the computer is in operation. Many small computers have the BASIC language in ROM—the machine "wakes up" ready to

execute commands in BASIC. Some computer manufacturers offer video games and other programs in plug-in ROM modules. This spares the user the chore of loading the program from a storage medium such as a cassette or floppy disk each time it is to be used.

Random access memory is used for storing transient information and instructions to the CPU. For example, while the BASIC language may be in ROM, BASIC programs are usually loaded from cassette or disk into RAM. When you're finished with one BASIC program, you can clear it from RAM and load another one in its place. Random access memory can also be used to store data, such as the values assigned to variables that are used in a program. Of course, RAM chips can store assembly language programs and programs written in high-level languages other than BASIC.

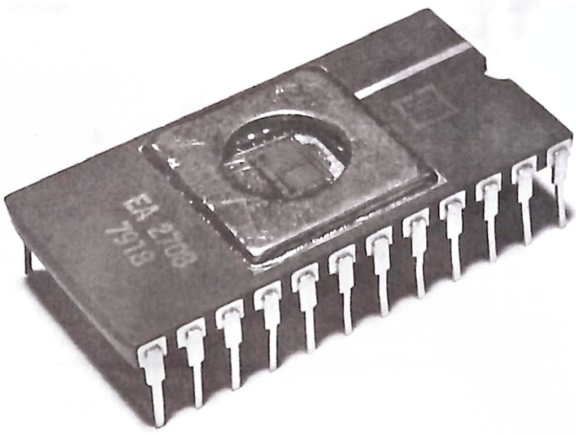
Two different technologies are commonly used for RAM chips. A *dynamic RAM* chip stores each bit of information in a tiny capacitor that must be recharged every few milliseconds to avoid loss of data. Each capacitor has an on (charged) state and an off (discharged) state. *Static RAM* chips use a tiny flip-flop circuit to store each bit of data. Once it's set to the on or off state, it stays that way until it is reset or the power is turned off. Static RAM chips are of three general types: CMOS, NMOS, and TTL. CMOS static RAM chips consume very little power, but they cannot be packaged very densely, so they are the most expensive for a given amount of memory. NMOS chips consume more power but can be made more dense than CMOS chips. TTL chips require the most power and cannot be packaged densely, but they offer the highest speeds of any type of RAM chip. Most static RAM chips use NMOS technology.

Because dynamic RAM technology is less expensive and lends itself to denser packaging than any type of static RAM, it is often used in microcomputer applications. But for all practical purposes, it really doesn't matter which type you have as long as it's configured correctly for your system. Only the power supply and your pocketbook really know the difference.

ROM chips fall into three basic categories: mask-programmed ROM, PROM (programmable read-only memory), and EPROM (erasable programmable read-only memory). A mask-programmed ROM chip is one that was originally manufactured with the programming in place. The manufacturer makes up a mask to a customer's specifications and then mass-produces the chips. This results in a high density chip, and it is inexpensive—but only if the ROM chips are purchased in a large quantity. The cost of making the mask is too high for short production runs.

PROM chips may be programmed in the field by "burning them" with a pulse of higher-than-normal voltage. The chip has many fuse-like links, and the appropriate pattern of gates that are on and off is established by literally destroying some of the fuses. PROM chips are relatively inexpensive for small-quantity production, but cannot be reprogrammed once they are burned. Also, PROMs are low-density chips.

EPROM chips, on the other hand, can be readily programmed and then erased and reprogrammed in the field. Like PROMs, EPROMs are programmed by applying a high-voltage pulse, but unlike PROMs, they can be erased with ultraviolet light. Then EPROMs can be programmed



This is an EPROM chip. Its dominant feature is the window through which ultraviolet light passes when the chip is being erased.

again with a voltage pulse. Once programmed, an EPROM behaves exactly like a ROM or PROM. Although an EPROM chip offers higher density than a PROM chip, the initial purchase price is higher. However, because it is reusable, it may cost less in the long run—especially in experimental applications. You'll find EPROM chips as well as ROM and PROM chips in many personal computers.

How much memory should your computer system have?

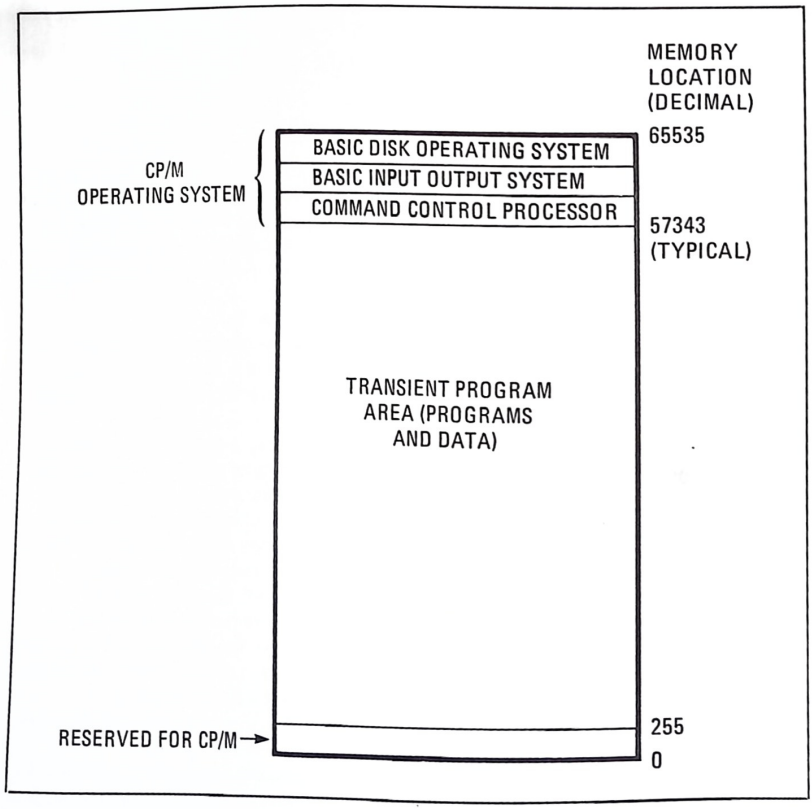
Memory chips are remarkably inexpensive these days. Enough dynamic RAM chips for 16K can be purchased for well below $20, an incredible figure when one remembers that 16 kilobytes of memory used to cost thousands of dollars.

When you consider the cost of the PC board itself and the support chips, the price for an add-on 16K RAM board for some systems may be $100 or more. Nevertheless, additional memory is one of the cheapest computer enhancements going. All of the popular 8-bit microprocessor chips are capable of addressing 65,536 one-byte memory locations. This is normally referred to as "64K" or 64 kilobytes, because 1024 bytes are considered to be one kilobyte. Some of this 64K must be allocated to ROM, but most personal computers have provisions for at least 48K of RAM—and it rarely makes much sense to have less than the full 48K. Many of the newer microcomputers employ *bank switching* so that considerably more than 64K of RAM is available. This technique allows an 8-bit computer to rival the memory capacity of the 16-bit systems, but at some sacrifice in speed.

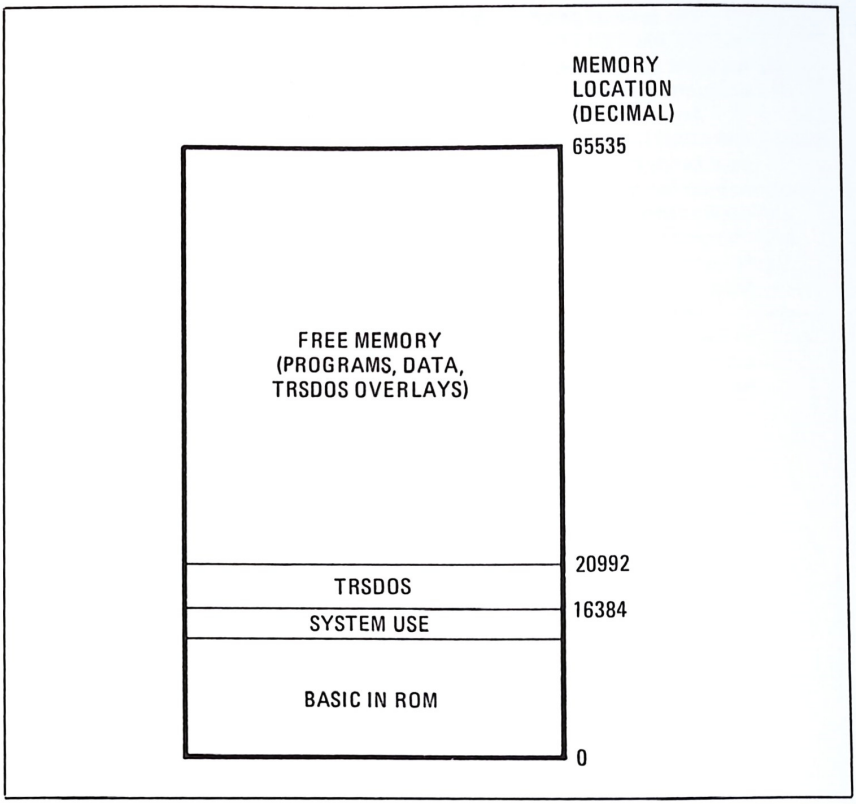
Another term you may encounter if you set out to add more RAM to your computer is *bank selectability.* On many RAM boards, the memory can be addressed in 4, 8, or 16K increments, which permits some of it to

be *deselected* to make room for ROM. For example, in most Z80-based CP/M systems the ROM is located at the highest addressable memory locations, often in the top 1K (i.e., between locations 64512 and 65536). As a result, you cannot have a full 64K of contiguous RAM—you must deselect some.

This may seem complicated, but if you examine the CP/M memory allocation map, perhaps it will be more clear. Each byte of memory has a specific location, and only one byte can be placed at a given location. The CP/M map shows the typical memory allocation in a Z80-based 64K computer using the CP/M operating system. Other types of computers must provide space for the same functions, although they often position them at different locations among the 65,536 possible addresses. The TRS-80 Model I and Model III have the ROM for BASIC and the operating system at the bottom of memory instead of the top, making it very difficult



Memory map of a 64K CP/M computer.



Memory map of the TRS-80 Model III.

to use normal CP/M even though these computers use the Z-80 microprocessor (see the TRS-80 memory map).

Bulk Storage Devices: Tape

So far, we've been talking about storing data and computer instructions (i.e., programs) in random access or read-only memory. This may be fine when the computer is running and only a relatively small amount of information needs to be stored, but there must also be a way to store larger quantities of information permanently. The answer is a *storage medium* such as magnetic tape or a floppy disk.

The least expensive means of bulk storage is audio tape. In this method, each bit of computer data is converted into an audio signal that is recorded on a standard audio cassette. Most inexpensive personal computers use this approach.

As anyone who has used it can tell you, cassette storage has pros and cons, but it does work. To load a program from cassette, you first command the computer to load the program. Then you play the tape, hoping the playback level is right. If several programs are on the tape, you have to locate the right one (or just play the whole tape and wait for the computer to pick out and load your chosen program). And if you should forget the exact name under which the program was originally saved on tape, you may never be able to retrieve it!

Saving a program on tape is also cumbersome. You tell the computer to save the program and quickly start the recorder running (before the computer starts sending data). If the recording level isn't right, you may have a problem. And if you forget to push down the "record" button along with the "play" button, you're out of luck.

Because not all cassette recorders are compatible, a program that was saved on tape with one recorder may or may not load when the tape is played back on another machine. Our Sinclair computer, for instance, will only save and reload programs dependably on one of about a dozen cassette recorders that we've tried.

Murphy's Law often applies here. If the recorder is going to malfunction, it's guaranteed that you won't discover it until after you've cleared the program from memory. Then you get to start all over again and rewrite (or at least retype) the entire program.

Another annoying trait of cassette storage is its agonizing slowness. A 20K program may take ten minutes to load on some computers. If you repeatedly revise and re-save a long program, you'll learn what a nuisance that can be.

Also, when you decide to modify a program you'll discover still another limitation of cassette storage: the new version will almost certainly be longer than the original, and that means it won't fit on the tape where the original one was. If you try to put it back, you'll wipe out the next program. The only safe way to store programs on cassette tape if you're going to modify them is to have only one program on each side.

Moreover, there are data storage and retrieval functions that aren't practical with a cassette system. For instance, some of the more sophisticated programs in this book require "data files," that is, separate files of information that the computer must be able to quickly retrieve and replace. Certain types of data files can sometimes be stored on cassette, but even then it's difficult and very inconvenient because the computer has no way to turn on the recorder and access the needed files.

Nevertheless, the cassette method is a widely used method of storing large quantities of data. Thousands—perhaps millions—of personal computer owners use this system. However, as a computer user's interest in the hobby grows, there comes a time to think about something better. These days the logical next step is floppy disk storage.

Floppy Disk Drives

Almost every personal computer manufacturer offers a floppy disk system, at least as an add-on accessory. The more expensive personal computers include disk drives as standard equipment.

A floppy disk is a round sheet of magnetic material that spins inside a protective envelope. The illustration shows a typical floppy disk. "Floppies," as they are sometimes called, come in two popular diameters, 5.25 and 8 inches. Most personal computers use the smaller size, while larger, business-oriented systems frequently use 8-inch disks—the kind originally introduced by IBM. Japanese manufacturers are increasingly

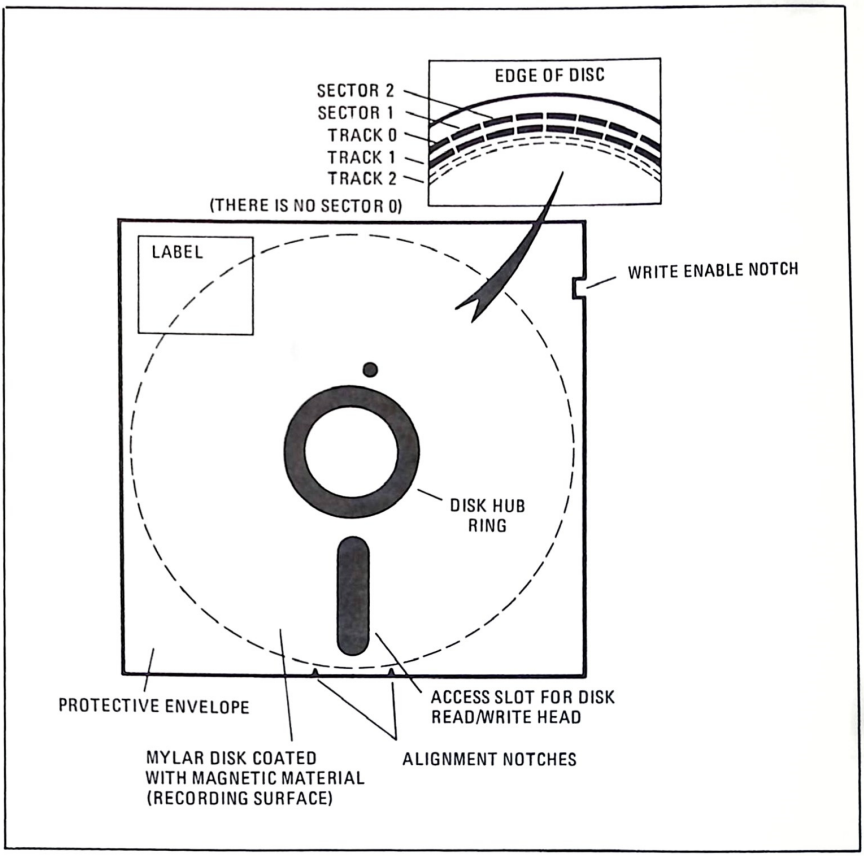


Diagram of a 5¼-inch floppy disk.

turning to a 3.5-inch floppy disk format. Despite the small size, these tiny disks store upwards of 400,000 bytes of data.

After using a cassette system, most people find a disk drive to be a revolutionary improvement in speed and convenience. A disk will save or load a 20K program in seconds rather than minutes, and the disk controller software automatically places the program wherever there may be room on the disk. Thus, if you revise a program and make it longer, there is no problem when you save it, unless you exceed the total storage capacity of the disk. The system simply finds blank space somewhere on the disk and puts part of your program there. A program need not be in sequential order in one place on the disk. The operator may not even realize all of this is going on inside the disk drive.

Another very attractive feature of a disk drive is its ability to access any part of the disk almost instantly. It doesn't have to run through all of the data on the disk to find the right file. In short, a disk drive is a random access storage device; a cassette tape system is a sequential access device.

Each floppy disk has a "directory" that tells the system exactly where all parts of a given program may be found, and the drive quickly moves to the right place to read or write the data. The directory also allows the user to quickly see exactly what's on any disk, complete with program sizes and the correct spelling of each name.

Still another advantage of floppy disks is that data can be copied from one to another very quickly, making it easy to "back up" (i.e., make a spare copy of) all important data so it can't be accidentally destroyed. Many computer users prefer to have two disk drives to facilitate copying programs from one disk to another for backup purposes. A typical disk with 100 kilobytes of data on it can be copied in just one or two minutes.

If floppy disk drives do all of these wonderful things, why don't all personal computers come with them? The main reason is cost: a typical 5.25-inch disk drive and its associated disk controller circuitry sell for several hundred dollars. Still, almost every computer user who has made the transition from cassette to floppy disks will tell you it's well worth the investment. A disk drive opens up a whole new world of programming possibilities.

This is not to say that floppy disks and disk drives are perfect. Far from it. If mishandled, a floppy disk can be so badly damaged that all the data on it will be lost. Even without mishandling, a floppy disk sometimes "crashes" for unexplained reasons—usually at the most inconvenient time possible. And disk drives themselves are not 100 percent dependable. A drive may slip out of alignment and refuse to read perfectly good disks. If that happens, the whole system will be down until the defective drive is repaired or replaced, unless the system has two or more drives. Even then, Murphy's Law dictates that the bad drive will be the one used to "boot up" (i.e., turn on) the system. Anyone who isn't prepared to wade in and switch drives at that point is pretty much stuck. But despite all of this, owners of disk-oriented personal computers who know what it's like in the world of cassette tapes almost universally swear by their floppy disks.

Floppy Disk Characteristics

Floppy disks have several important characteristics, as shown in the illustration. A disk has *concentric tracks,* each one divided into *sectors.* Typical 5.25-inch disks have 35 or 40 circular tracks on which data is stored, but some high-density drives write as many as 77 tracks on a 5.25-inch disk. Each track is divided into a specific number of sectors (often 10, 16 or 18), with each sector containing a specific amount of data (typically 128 to 512 bytes).

The computer's software tells the disk drive's *read/write head* to move to a particular track to read or write data, one sector at a time. A stepping motor moves the head to the right track. Then the head must read or write the correct sectors. The drive determines where one sector ends and the next one begins by a counting procedure. On *hard sectored* disks there are either 10 or 16 sector holes (depending on the type of disk drive system in use), and these mark the sector boundaries. *Soft sectored* disks have just one hole that marks the beginning of each rotation, and the disk controller software determines where each sector begins and ends.

This business about tracks and sectors brings us to another important point about floppy disk systems. Data can be placed on a disk in a very dense or very sparse format. In a *single density* system, the data is relatively uncrowded on the disk. *Double density* systems place approximately twice as much data on the same disk, and use sufficiently precise data seeking, reading and writing functions to do so accurately. Some computer manufacturers offer *quad density* systems; these often use disk drives with two heads, one to read and write on each side of the floppy disks. In fact, a few manufacturers even offer *octal density* with four times the normal recording density on each side of a two-sided system.

Once you begin making serious use of your computer, there never seems to be enough data storage capacity on any floppy disk, so systems offering more than single density are obviously desirable—but also more expensive.

A typical 5.25-inch single density, single-sided floppy disk will store between 80K and 90K of data, depending on the particular format in use. How much is 90K? For text matter, that is equivalent to 90,000 characters. For example, the second and third chapters of this book occupy about 90K of disk space.

Double density 5.25-inch disks typically hold about 160K, and quad density disks will store upwards of 350 kilobytes of information. For comparison, a single density, single-sided 8-inch floppy disk will store about 240K, while the double density version will store 500K, or half a megabyte of data.

In business-oriented systems, it is not unusual to have two megabytes of on-line floppy disk storage, coupled with a *hard disk* system that can store 20 to 200 megabytes of data. A hard disk system is considerably more expensive than a system built around floppy disks, but for a serious business application the cost is often justified. A hard disk system is not only capable of storing vast quantities of data, but it also can access it in a fraction of the time required by a floppy disk drive.

Hard-Copy Printers

Once you have a computer and some form of mass data storage, the next acquisition is usually a printer. There are several popular types of printers, and they are interfaced to computers in two different ways that we'll explain shortly.

For most computer hobbyists, the most practical printer is the *dot-matrix* type, which forms characters from patterns of dots. Its print head has pins that strike the paper to produce the dots, and the printer's logic circuits activate the right pins to form the dot pattern for each character.

Dot-matrix printers are capable of very high speeds. Modestly priced models print 80 or even 160 characters per second, and many print bidirectionally—that is, when the head is at the right side of the page after printing a line, the next line is printed from right to left to eliminate the usual delay while the head returns to the left side of the page.

These printers usually have *tractor feed* mechanisms to keep the paper properly aligned during long printouts. If you ever print a long document without fan-folded paper and a tractor feed system, you will know how tedious that can be.

A dependable dot-matrix printer only costs a few hundred dollars. However, one disadvantage of the lowest-priced dot-matrix printers is that they only accommodate paper up to about 9½ inches wide, which precludes the use of the widest computer forms. Most manufacturers also offer printers capable of handling paper as much as 15 inches wide, but at a higher price.

The major drawback of dot matrix printers is their "computer print"—even a layman can tell that the printout came from a computer. In many applications this doesn't matter; it may even be a virtue. But if you're interested in word processing applications, the dot-matrix typestyle may be objectionable. On the other hand, some of the newer models offer two printing modes, a high-speed mode and a slower mode with much better printing quality.

Nevertheless, those who use personal computers for word processing tend to purchase letter-quality *daisy-wheel* printers. These printers are much more expensive: many models still sell for healthy four-figure prices. They are also slower than dot-matrix printers. However, they do produce type that closely resembles the output of an office-quality electric typewriter. A wide variety of typefaces may be selected, and either a film ribbon (for formal correspondence) or a reusable nylon ribbon (for less important work) may be used.

The daisy-wheel printer gets its name from the shape of its type wheel, which actually resembles a daisy—but with many extra petals. To print, the machine performs a number of sophisticated functions very quickly. First, the print head moves across the paper rapidly, abruptly stopping at exactly the right place for each character. Then the type wheel quickly rotates to the proper position for the desired character and a solenoid forces the character to strike the paper. That such printers are as fast as they are (up to 55 characters per second for some models) is amazing.

Among the most popular brands of daisy-wheel printers are the Diablo and the Qume (both made in the United States), and the NEC Spinwriter, made in Japan by Nippon Electric Company. Technically, the Spinwriter should be called a "thimble" printer rather than a daisy-wheel printer because of the unique shape of its type wheel, but it works on the same principle as a daisy-wheel printer, and it has an excellent reputation for dependability among word processing system owners.

There are also other types of printers available. Older Selectric-based computer terminals and teletypewriter units are plentiful on the surplus market. For the computer hobbyist who doesn't need high-speed printing—and is willing to repair a well-used print mechanism—one of these may be an excellent choice.

Electrostatic and thermal printers are other alternatives. Several inexpensive units of these types are still sold by various manufacturers, but because they require special paper and produce poorer-quality print than the other kinds of printers, they are declining in popularity.

Parallel and Serial Ports

Something should be said here about the two major formats used by computers to communicate with such peripherals as printers, telephone modems, and video display terminals. To perform useful work, computers must exchange large quantities of data with these devices. For instance, a computer must send a stream of data to a printer before there can be a hard-copy printout. And a computer must have a two-way data exchange with a modem to communicate with another computer at a remote location by telephone.

As we said earlier, the popular 8-bit CPU chips have eight side-by-side data lines, each carrying one bit of each byte that the CPU sends or receives. All eight bits are sent or received *simultaneously*. This is an example of *parallel* data exchange. Many personal computers use this parallel arrangement to send and receive data from external devices, especially printers. Connectors with 15 to 50 pins are used for parallel interfaces.

Why use a connector with so many pins for just eight single-bit data lines? In addition to the data lines and a ground wire, the parallel port must have a *data strobe* line to tell the receiving device when each byte is ready to be received and an *acknowledge* line to tell the sending device that the receiving device got the last byte sent.

Because the CPU itself sends and receives data in a parallel fashion, the parallel arrangement is the least expensive way to exchange data between a computer and its peripherals. It is also the fastest method of transferring data, so it is used for data exchange between a computer and its disk drives. An increasing number of personal computer owners use a parallel interface to deliver data to their printers as well. Many printer manufacturers have standardized on this interfacing method, offering a *serial* interface only as an extra-cost accessory.

However, parallel interfacing does not lend itself to efficient long-distance communication between computers because it requires so many different lines (or radio channels) and because of compatibility problems. For efficient communication, the serial method of data exchange is a better choice. And if a computer has only one peripheral interface, a serial one may be preferable because it allows both computer-to-computer data exchange and printer interfacing.

For serial communication to be possible, the parallel data must be converted into a sequential stream of bits. A *USART* (universal synchronous/asynchronous receiver/transmitter) or *UART* (asynchronous only) chip performs this conversion. Each character to be sent or received is *framed* with a *start bit* in front of the data bits and one or more *stop bits* following the data bits.

The number of bits that are sent and received for each character varies somewhat. There may be anywhere between five and eight *data bits*, sometimes with a *parity bit* for error checking, plus one or two stop bits. The essence of serial data exchange is the sequential one-bit-at-a-time manner of communication. It permits data transfer on a single channel or line, although a second channel in the return direction is often needed for two-way communication or *handshaking*. In typical modem applications, of course, the two channels consist of two audio frequencies separated by about 1000 Hz, both using the same telephone circuit. The newer high-speed modems take this one step further, with four frequencies used to send data in each direction simultaneously.

Handshaking and the RS-232 "Standard"

Handshaking is the process by which the sending device tells the receiving device it is ready to send data, and by which the receiving device tells the sending device it is ready to receive it. There are several popular methods of doing this. Some involve only software, typically a method of sending a predetermined character from the receiving end to tell the sending device to begin transmitting, with another character to tell the sending device to stop transmitting. Another common way to accomplish the same thing with hardware is by using a separate line running between the sending and receiving devices. The line can be configured so that a logical one (typically a positive voltage) means to start sending and a logical zero (a negative voltage) means to stop sending.

Whatever handshaking or *communications protocol* arrangement is selected, the essential requirement is for the sending and receiving devices to be compatible. Each must understand the meaning of the start/stop signals sent by the other.

The sending and receiving devices must also be compatible in their transmission speed, or *baud rate*. No meaningful data exchange can occur unless the two devices send and receive data at the same speed. The speeds are sometimes set by jumpers or tiny DIP switches on the serial interface board, but they may also be controlled by software. The most common data rates for modems are 300 and 1200 baud, with the faster

rate requiring the four-frequency arrangement mentioned earlier. A speed of 1200 baud is roughly equivalent to 120 characters per second.

Speeds of 300 to 1200 baud were once standard for printers, too, but recently it has become more common to send serial data to a printer at a much higher speed, with the printer storing the data in a buffer of on-board memory. Since few low-priced printers are fast enough to keep up with the incoming data stream even at 1200 baud, a buffer is required to make efficient use of higher speeds. Without a buffer, the computer will still send each character quickly, but then it must respond to a handshaking signal and wait until the printer is ready for the next one. The advantage of high-speed data transmission and buffering at the printer, of course, is that it more quickly frees the computer to do other work.

For communication to occur, the sending and receiving devices must also speak the same language, as it were. The ASCII code (American Standard Code for Information Interchange) is the standard, but other codes (some of them binary) are also used at times. In teletypewriter communication Baudot has long been the standard.

Serial communication is usually based on the industry-standard *RS-232C* interfacing arrangement. This standard specifies a 25-pin connector for serial interconnections, and also sets forth what type of signal goes on each of the pins. Most serial interface boards have documentation that tells—or at least implies—what each pin does. However, keeping track of which end of the line does what can be confusing. For example, a serial printer port on a computer should be configured so the transmitting line is pin 3 and the receiving line is pin 2. But from a printer's or terminal's viewpoint, the transmitting line has to be pin 2 and the receiving line pin 3.

To further complicate matters, a modem usually has the same send-receive configuration as a computer's printer port, which means the wires going to pins 2 and 3 often have to be reversed for the two devices to be able to communicate. A good solution is to make up a cable with a double-pole, double-throw (DPDT) switch to reverse pins 2 and 3.

Unfortunately, "standard" RS-232 interfaces are by no means standard. Many printers use pin 20 (called *data terminal ready*, or DTR) for a handshaking voltage, but others use any of several other possible choices—including pins not even mentioned as proper places for handshaking signals in the IEEE standards for RS-232 interfaces.

Often it is necessary to jumper two or three specific pins before your computer and printer or modem will actually talk. Many computers will not send any data over an RS-232 line until the proper jumpering is in place. Even worse, some computers simply "hang up" and refuse to do anything if they encounter a print command at a time when the RS-232 port is unable or unwilling to send data. If you're trying to debug a program that seems to hang up at a certain point every time, don't overlook the possibility that the computer thinks the printer isn't ready to receive data. (You did remember to turn the printer on, didn't you?) Getting serial RS-232 printers, modems, serial video terminals, and computers all talking to each other can be a difficult challenge at times.

All of this may sound absurdly convoluted, and in fact there must be a simpler way to do these things. But we offer one consolation: you won't

be likely to damage anything by miswiring an RS-232 cable—it just won't work if it isn't right. If you've been a radio amateur for long, you'll probably appreciate that nice feature: remember what happened the time you put 800 volts on the filaments of your expensive new vacuum-tube transceiver by miswiring two pins on the power supply connector?

Fortunately, with RS-232 connectors the compatibility problem can often be solved by trial and error, with the aid of a voltmeter. If the first pin configuration you try doesn't work, try another. Assuming you have the receive and transmit lines right, if two devices still won't communicate the next thing to check is the handshaking and communications protocol arrangements. If the sending device doesn't see the right voltages or handshaking signals, it won't send any data.

The RFI Problem

For a radio amateur, an especially troubling thing about some personal computers is their ability to act as *transmitters* or even as *receivers*. Radio frequency interference (RFI) can be a major problem, particularly with computers manufactured before the FCC established RFI standards for personal computers in 1981.

Some of the pre-1981 personal computers produced so much RFI that it was virtually impossible to use a receiver in the same house with one. The problem today is much less serious, but some attention to the RFI problem is still necessary. A computer is, after all, an RF device as well as a digital one: its main clock typically operates in the HF radio spectrum. The clock signal is not only strong and broadbanded, but it is also rich in harmonic energy.

For pre-1981 systems, wholesale shielding is in order if the computer and radio station are to be used simultaneously. Many of the earlier computers came in nonmetallic boxes and used unshielded cables for all of their peripheral interconnections. A number of articles have been published in radio magazines describing the modifications required for various pre-1981 computers.

For most such systems, the first step is to completely enclose the electronics in a shielded box. This could mean eliminating the original cabinet entirely, and may involve considerable work. In addition, all of the interconnecting cables may have to be shielded, and the AC power line may require extensive filtering to suppress RF at many different frequencies. Often the best approach is to do this one step at a time, only going as far as necessary to get RF emissions from the computer down to an acceptable level.

In newer systems, the problem is usually less severe. To meet the FCC standards, personal computer enclosures now have integral shields, often a screen imbedded in the computer's nonmetallic case. If you have such a computer, your receiver may still detect some computer-generated noise at certain frequencies. The problem can be minimized (or perhaps eliminated entirely) by placing the computer on a separate AC circuit from

the ham gear, and by experimentally moving the computer and rigs around the room.

Another thing that helps considerably is using a good, high antenna, with a well-shielded feedline down to the hamshack. Some computer users have solved difficult interference problems by eliminating a roof-mounted dipole and replacing it with a higher beam antenna or by moving the dipole further away from the computer. This is another illustration of the old rule—the higher, the better.

If an RFI problem persists even with a new computer, some additional shielding may be necessary. Ribbon cables to disk drives and printers may have to be shielded. Also, cables to such devices as joysticks are notorious RF emitters. If you're not using a particular peripheral, disconnect it. A hefty RF choke on the AC line—mounted at the computer and grounded to it—may be needed in difficult cases. Grounding each unit's chassis to the others may also help.

All of these steps may still not be enough. At the computer's crystal oscillator frequency (and at harmonically related frequencies), a strong signal will probably be present no matter what. For example, both the Apple II and Commodore computers emit S9-plus carriers on 7159 kHz and 14318 kHz—right in the 40 and 20 meter bands.

Even if the computer's RF is successfully trapped inside the computer, there may be an RFI problem going the other way: a computer—and especially the video monitor—may also be on the receiving end of RFI from a high-powered transmitter. When a video monitor has a television interference (TVI) problem, the conventional cures described in various texts are usually sufficient.

In short, the RFI problem can usually be reduced to an acceptable level if you're willing to go to enough trouble. And in many instances, there will be little or no problem. The authors have used a number of unmodified post-1981 computers in their hamshacks, and the interference problem has been only minor on most frequencies all the way from 1.8 to 1296 MHz. Granted, we use high tower-mounted antennas, but other amateurs report only minor problems with lower antennas. Merely repositioning the computer and isolating the AC line is often sufficient to eliminate the problem except on the computer's main clock frequency and its harmonics. Even there, the problem should be kept in perspective: our Apple computers may be 10 dB. over S9 at 7159 kHz, but Radio Moscow is often 40 dB. over S9 on an adjacent channel—and we certainly derive more benefit from an Apple II than we do from Radio Moscow.

Conclusion

Hundreds of pages could be written about the topics we've highlighted in this chapter. Because this book is primarily about software for ham radio, this is intended to be no more than a brief introduction to personal computer hardware. If you have a special interest in any aspect of this, you'll want to consult some of the many books that have been written about microcomputer hardware and interfacing problems.

CHAPTER

4

THINGS YOUR MOTHER (AND MAYBE YOUR DEALER) NEVER TOLD YOU ABOUT MICROCOMPUTERS . . .

As we developed the programs for this book, we learned a lot of practical things about microcomputers—things that it would have been nice to know sooner. It's an old cliche, but if we'd known what we know now when we started out, we could have saved ourselves a lot of time, trouble, and money—especially money. If you're fairly new to personal computing, maybe you can benefit from some of the things we learned. If you're an old-timer in microcomputers (that is, somebody who owned one before 1984), perhaps you can relate to some of what we say here. You may have had similar experiences.

Like Radios, Computers Proliferate

One inevitable fact of life is that computers are addicting. Every radio amateur is already involved in one technical hobby; such a person is a prime candidate for addiction to another.

Remember when you got your first piece of ham gear? Maybe it was just a radio receiver—but one capable of tuning something beyond the AM and FM broadcast bands. Soon you began studying and earned your first license. Then, of course, you got a transmitter or a transceiver.

That meant you had to have an antenna, right? So up went a wire dipole, maybe on a pole or just in the attic. Then you got on the air. Pretty soon you discovered that there's only so much a ham can do with just one little rig and antenna. If you're typical, you added another rig, maybe for

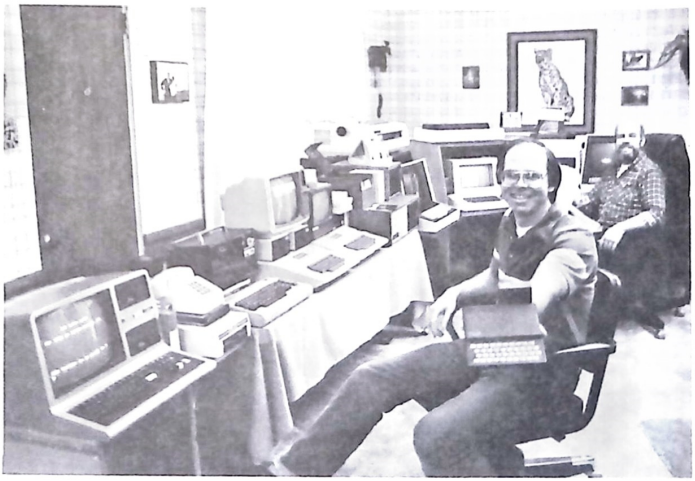
two-meter FM. Or perhaps you went the other way, starting on two meters and only moving to the HF bands once you got your general class license.

But it probably didn't end there. Pretty soon you knew you needed a tower and a beam. So up they went—and you first experienced that chronic disease of amateur radio: neighborhood animosity. People up and down the block complained loudly about TVI and started circulating petitions. By then you were not only using a beam antenna, but also a shiny new kilowatt power amplifier, right?

Sound familiar? It happened to most of us as our interest in ham radio grew. If you're new to computing, it's time to warn you (provided you've safely hidden this from your family): computers affect people the same way. Oh, there may not be any neighborhood petitions. An avid computer hobbyist can madly pursue his pastime until all hours of the night without the neighbors even knowing about it. But your family will know. You'll know. And so will your checkbook.

Now we know about this problem, but we didn't at first. Your authors began with one computer—a Northstar Horizon that was purchased mainly for word processing. But by the time this book was finished, we owned nine computers: three Apples (with a jillion Apple accessories), a TRS-80 Model III, a Xerox 820, a Commodore 64, a VIC-20, a Sinclair ZX-81, and the original Northstar. These systems included a total of 17 disk drives. Oh, we also ended up with three printers, including a fancy daisy-wheel job. And there were three modems, and . . . well, you get the idea. But it all seemed so innocent and harmless at first.

The most successful personal computer companies understand the addictive qualities of their products. They know human nature, and they exploit it in their marketing. Sure, some of them will fix you up with their basic computer for a ridiculously low price. But they know darned well



You may start out with a little computer like the one co-author KC6A has in his hand, but if you're a red-blooded radio amateur, it probably won't be long before you have a roomful of computer hardware. If you look carefully, you'll see (from left) a TRS-80, Hayes Smartmodem, Commodore 64, two Apple IIs, a Northstar Horizon IV, a Xerox 820, two CRT terminals, a NEC Spinwriter, and two dot-matrix printers (one partially hidden, beside N6NB).

that you'll be back for the rest of the system once you're hooked. According to *Business Week,* some of the largest personal computer manufacturers gross more from the sale of add-on peripherals than from the sale of the computers themselves.

If you started out with a stripped-down personal computer, you may know already just how true this is—and how costly it can be to upgrade your system one piece at a time. Probably the first thing that happens to many new computer owners is that they discover terrific programs they'd love to run, if only they had enough memory or disk storage capacity. Without disk drives and at least 48K of free memory, your computer can't do some of the most interesting and useful things that microcomputers are capable of doing.

Some computer hobbyists resist for a while, but eventually the "I've-gotta-have-a-beam" syndrome usually sets in. The family may be wearing last year's shoes, but a disk drive miraculously appears. And so does a printer. After all, without a hard copy printout of your best programs, what do you have? Just memories of fun and games, that's what!

Now there's really nothing wrong with all of this. Provided other things (like job and family) don't suffer, why not invest a little time and money in a rewarding hobby? And why not invest enough to do it right? Why waste your valuable time struggling along with a unit that feels like a mere toy when you could use your time much more productively if you had hardware good enough to do the job right?

Fair enough. But if you're just getting started, you should understand what you're getting into, and what the total package will cost. And if you're sure you won't get into computing that seriously, you should know the limitations of that bargain-basement computer before you buy it.

A Friend's (or Dealer's) "Perfect Computer" May Not Be

If you're thinking of buying a computer, you probably know several people who already own one. Goodness knows almost everybody you talk to on any ham band these days is at least somewhat interested in computers.

But if you're looking for advice about computers, you may notice that computer owners are a loyal bunch. Now brand loyalty isn't unique to computer owners, but the computer industry has enjoyed far more than it deserves. Computers are mysterious gadgets, and they're frightening to many people. Once someone struggles through the initial learning process and masters one computer, he or she tends to be a bit biased about that brand.

This certainly isn't true of everyone, but some computer hobbyists are strictly one-brand types. They know their own computers, and they'd rather not reveal how much they don't know about the others. So they pretend the others don't exist—or shouldn't exist.

As we prepared the programs for this book, we were forced out of our own computer myopia. We started out with an S-100 bus computer running CP/M. And we'll admit it: we shared some of the typical S-100

owner's attitudes toward less elaborate personal computers. Then we bought our first Apple—just to make sure our programs would work on one, you understand. Our attitude quickly changed. Every computer has its pros and cons, but the Apple introduced us to a whole new world of software availability, moderately priced accessories, and just plain *fun.*

A little later, we bought the TRS-80, also to make sure the programs in this book were compatible with that popular brand. And again, it was an enlightening experience. We quickly discovered that the TRS-80 also has some real advantages.

But as we learned about the various brands of computers, we kept running into people who thought their own brand of computer was the only one worth knowing about. We know a fellow who's an Apple person with a capital A. He probably wouldn't know an IBM PC from a Sinclair. He thinks a Xerox 820 is a photocopying machine. When we broke the news to him that we'd bought a TRS-80, the reaction was interesting.

"Oh. Is that the one that looks like it should be a terminal?" he responded, nose high in the air. If we'd just told a Mercedes owner about buying a Plymouth, we couldn't have gotten a snootier response. But the funny thing is that we also know TRS-80 owners who think *their* computer is the Mercedes and *Apples* are the economy models.

Our point: the fact that a particular computer may be perfect for your best friend (or local dealer) may not necessarily mean it's best for everyone. If you already have a favorite computer, please be patient while we tell the folks who own other brands how to make our programs work on their systems. And if you're about to buy a computer, check them all before you write a check.

Computer hobbyists tend to choose up sides, becoming ardent fans of their own brands. This Apple enthusiast doesn't mind telling the world how he thinks Apples compare with Radio Shack TRS-80s. Thousands of TRS-80 owners would take exception to this bumper sticker, of course.

Incompetence Is Rampant in This Business

The microcomputer industry has grown from nothing to a multibillion dollar business serving millions of customers in only a few years. Growing pains are inevitable under those circumstances.

One of the worst of those growing pains is the shortage of qualified sales and service personnel in the industry. Sadly, most computer stores have been forced to hire salespeople who know a lot more about how to play video games than they do about the technical questions an experienced radio amateur is likely to ask. Sometimes the best thing a shopper can do is to hope the salespeople are busy with other customers and get a hold of the catalogs and manuals. The documentation may not be all that great either, but at least it was probably written by someone who knew something about the product.

But let's not be too hard on the salespeople. The fact is that they're expected to know everything about every system. And they're so swamped with customers asking dumb questions that they couldn't possibly keep up with their reading. They just don't have time to read all the catalogs and instruction manuals (or even to test all of the software packages) that come into their stores.

In the customer-service end of things, it's even worse. The people in that field are absolutely overwhelmed. They have to deal with everything from knowledgeable customers with serious questions to absolute idiots who know nothing but think they know everything. In our experience, about 90 percent of all computer service problems stem from user mistakes, not flaws in either the hardware or the software.

But the trouble is that the service types are so busy rushing from one phone call or irate customer to the next that they can't tell legitimate problems from cockpit errors. And they may not have the time to keep up on their own products, either.

We recently spent an hour watching the man in charge of customer service for a modest-sized OEM (original equipment manufacturer), a company with about 75 dealers around the world—and a product line built around after-market upgrades for a popular personal computer.

This service manager had half a dozen walk-in customers in rapid succession, as well as 20 phone calls. At any given moment, he was juggling a couple of visitors and several people on the phone—all of them demanding solutions to problems. Looking anything up was out of the question: he had no choice but to wing it, answering questions about everything from how to indent paragraphs with a word processor to ways of initializing an RS-232 port to make it handshake properly with a serial printer.

During the hour we were there, we caught this harried service rep giving clearly incorrect answers on at least three occasions. But even if he was aware of these mistakes, he couldn't correct himself. He had to keep rushing from customer to customer, trying to keep up with the blinking "hold" lights on his phone. Never mind that some of the callers would probably waste a lot of time because his answers led them astray. There just wasn't time to conduct research for each caller or visitor.

Most troubling of all, this man says we witnessed a typical hour on a typical day. And we suspect his experiences are repeated daily throughout the microcomputer industry.

As you might imagine, we didn't get six different brands of computers up and running without discovering the problems of customer support firsthand. We can tell plenty of war stories about OEMs and dealers who didn't have the facts about their own products—and really didn't care. We'll give two examples.

In the first case, we found a bug in the firmware for a popular Apple II serial interface card, a bug that made it impossible to align columns of tabular matter on a printer as we could with other serial interfaces. We contacted the manufacturer, expecting not only to be heard, but also that a fix might be forthcoming. We were dreaming. Someone who seemed technically competent listened for a few minutes, but then he "solved" the problem by abruptly hanging up on us and refusing to take our calls thereafter. His customer service philosophy was pure ostrich.

Some manufacturers take the ostrich approach a step further: they won't even accept calls from end users of their products. We handled an institutional purchase of three mainframe-type personal computers a while back. It became clear right away that all three computers had a problem in seeking the correct track and sector during certain floppy disk read operations. As a result, text files and programs often came out badly garbled.

We called the manufacturer, hoping there was a solution to what was obviously a common bug in this well-known computer. But no one at the factory would even talk to us. "See your dealer," they said. Never mind that the primary dealer in the area had moved 100 miles away, and that the others had such a minimal knowledge of the product line that they couldn't begin to understand the problem, much less help solve it.

We eventually ran across an engineer at the factory who had witnessed the same disk-reading glitch himself, but he had neither the time nor the interest to try to fix it. His assignment was to develop new products, not fix the old ones. To this day, the three computers are flaky during certain disk operations, and the institution has responded in the only way it could: by blacklisting the manufacturer.

If you've been around microcomputers for long, you probably have similar stories to tell. If not, be thankful. But above all, if you find a customer service person who is technically competent—and who has the time to answer your questions—buy him booze, baseball tickets, or whatever it takes to keep him on your side!

There Are No Free Lunches—Or Are There?

If customer service is as marginal as it seems to be in much of the microcomputer industry today, some radio amateurs and other technically oriented computer hobbyists wonder why they shouldn't just buy the product at the lowest possible price—and neither expect nor pay for any customer support.

Hams have an old tradition of fixing their own hardware, and of helping each other solve technical problems. We think that tradition is alive and well in the amateur radio/computer hobby. So why not just look at the ads in the computer magazines, and buy the product at the lowest possible price—by mail order?

Many hams do precisely that. One can save hundreds of dollars on most computers by shopping carefully, and by not being afraid to wade in and do one's own assembly and system integration. For example, it's possible to buy a stripped-down Apple or TRS-80 at a fraction of the system price, and then shop for below-retail prices on disk drives and other accessories. Apple has taken steps to prevent mail-order sales of its own products, but independent OEMs and dealers do a brisk mail-order business in Apple-compatible accessories.

That's fine, and we believe in it. But don't expect the accessory to come with documentation that tells you how the pieces all fit together, because it may not. We have repeatedly come across documentation that said quite clearly to hook something up a certain way, when it turned out the correct configuration required exactly the opposite polarity or pin configuration. You may trade hours of anguish and frustration for the dollars you save. If you really know what you're doing (or, best of all, if you know someone who has already done the same job), you really can get the proverbial free lunch with mail-order add-ons. But you may also buy yourself a lot of troubles, and you run the risk that the whole thing may never work the way it was supposed to.

The Horsepower Race: Memory and Speed

Older readers will remember the bizarre horsepower race of the 1950s, when each Detroit car-maker tried to outdo the others by offering "muscle cars" with more and more horsepower. From a standing start, some of these cars could exceed any speed limit in the land within 10 seconds. So what if they only went a few miles on each gallon of gasoline? Who cared? Gas was 30 cents a gallon and we had plenty of it.

In the microcomputer industry, there's a new kind of horsepower race. It doesn't encourage people to break the law or waste gasoline, but it's a race nonetheless. This one involves memory and processing speed.

While you should get as much memory and computing speed as possible for your money, things should be kept in perspective. For example, what you need is *usable* memory. When Commodore introduced its 64K computer, there was considerable hoopla about the fact that this economical computer offered 16K more memory than several more expensive Apple and Radio Shack models then available.

When we were shopping for our Commodore 64, salespeople repeatedly told us how much more memory than an Apple II it had. However, when you deduct the memory set aside for the system ROM package, it turns out that the Commodore 64 offers only about 39K of memory for BASIC programs, virtually the same amount as is available in a 48K Apple II and TRS-80. We're not saying the Commodore 64 isn't a

good buy. It's just that it doesn't offer significantly more usable memory for BASIC programming than its 48K competitors, the chorus of false claims from salespeople notwithstanding.

For that matter, some of the Apple look-alikes (such as the Franklin Ace 1000) for a time engaged in the same sort of RAM one-upmanship—also without offering more memory that is really usable for BASIC programming applications.

Apple eventually joined this horsepower race. The Apple IIe comes with 64K of RAM built in, even though the extra 16K cannot ordinarily be used for Applesoft BASIC programs. And Apple went a step further, offering an accessory board with another 64K of bank-switchable memory—which is also unavailable for most BASIC programming applications. Having this extra RAM must be a little like driving a 400-horsepower car around town: it feels good to know it's there, even if you really can't use it very often.

The other horsepower race involves computing speed. People who wouldn't have known a megaHertz from a megavitamin a few years ago are bragging that their systems run at "2 megaHertz," "4 megaHertz," or even faster. Again, a fast CPU is generally better than a slow one. But a higher clock speed doesn't always translate directly into greater real speed in handling programs or data. There are, in fact, significant differences in real processing speed among the various popular computers, but they don't exactly correlate to clock speed.

To illustrate those differences, we conducted an experiment with some of the programs in this book—the ones that make the greatest demands on a computer.

For pure number-crunching, the program that computes azimuth and elevation coordinates for the moon has no equal in this book. Check out the formulas in the program listings on pages 310-321 to see what we mean. We ran the moontracking program on several computers so we could compare processing speeds on this kind of program; the following table shows the results. In each case, we had the computer calculate lunar azimuth-elevation data every 60 minutes, over a two-month time period starting January 1. We let all of the computers run for exactly 20 minutes and noted how many days and hours into the year each one got. Then we calculated how long it took, on the average, for each system to compute the moontracking data for one day.

The information in this table requires some explanation; it could be misunderstood. We tried an Apple II in three configurations: the standard version running Applesoft BASIC, a Z80-based Apple with a "Softcard" in accessory slot 7, and the standard version again, but with the moontracking program compiled.

It would take us a bit afield to explain how a BASIC compiler works. Suffice it to say that it creates a machine-language version of a BASIC program that a computer can directly execute. The result is an increase in program speed, often a dramatic one.

We have compiled many of the programs in this book using the Hayden Applesoft Compiler. MOONTRACKER is about 70 percent faster when compiled. That's a very worthwhile improvement (it makes an Apple

Comparison of time required to compute moontracking data for one day on various computers

TYPE COMPUTER	CLOCK SPEED	TIME REQ'D TO PROCESS MOONTRACKING DATA (ONE DAY)
Northstar Horizon	4.000 MHz	31.14 sec.
IBM PC	4.770 MHz	32.10 sec.
Xerox 820-II	4.000 MHz	34.00 sec.
Apple (compiled)[1]	1.023 MHz	38.20 sec.
Xerox 820[2]	2.500 MHz	49.06 sec.
Apple CP/M (Z80A)[3]	2.041 MHz	58.79 sec.
Standard Apple II[4]	1.023 MHz	70.07 sec.
Commodore 64	1.020 MHz	78.14 sec.
ZX-81 (fast mode)	—	105.00 sec.
TRS-80 III	2.028 MHz	149.00 sec.

[1]This is an Apple II in its normal 6502 configuration, but with the moontracking program compiled using the Hayden Applesoft Compiler.
[2]This is the original version of the Xerox 820, as opposed to the newer 4 MHz version.
[3]This is an Apple II equipped with a Z80 softcard, running Microsoft BASIC rather than Applesoft BASIC.
[4]This is a normal Apple II/IIe running Applesoft BASIC.

do its work almost as rapidly as a more powerful computer), but it's nothing compared to the improvement in speed we saw on some of the other programs. Our GRAYLINE program (which computes sunrise/sunset times for more than 400 locations around the world, compares them with local sunrise/sunset times, and sorts the results) executes three times as fast after being compiled. In our test configuration, the uncompiled version of that program took 12 minutes to run on the standard Apple II (and more than nine minutes even on the Northstar), but it ran on the Apple in just over four minutes after compiling (including the time it took to print a hard copy)!

Execution speed is all-important in contest operating software. We went to great lengths to make our DUPECHECKER and LOGGER programs execute rapidly. They're fast programs under any circumstances (they will search through 1000 calls in memory and determine if a particular call is a repeat in about one second), but the compiled version of the DUPECHECKER is awesome in its response speed.

To increase a computer's execution speed when running BASIC programs, a compiler is a great acquisition. Unfortunately, all that added performance is not without some cost. The compiled versions of most programs are at least twice as big as the uncompiled versions. For a program that stretches the limits of a computer's usable memory, that rules out compiling. The logger programs cannot be compiled and run on a typical 48K or 64K computer. And the DUPECHECKER, which normally

will store more than 4,500 call signs on an Apple II, can only handle about 1,800 after it is compiled.

Another point we should emphasize about the execution speed test summarized in the table is that it called upon the computers to perform complex mathematical functions, not the more typical searching and sorting done by many computer programs. The TRS-80 was disappointingly slow in running the moontracking program, but its speed is much closer to that of the other computers in running other types of programs. And it has other features that we like very much, some of which are explained in the next chapter. Execution speed may be an important kind of computer "horsepower," but it is only one of many considerations in selecting a computer.

Murphy's Law and Microcomputers: AC Line Troubles

Radio amateurs quickly learn about Murphy's Law ("If anything can go wrong, it will"). It applies equally to computers, large and small.

Every experienced computer user has horror stories about Murphy's Law. Some of the worst involve the AC power that quietly runs around in your walls and that you normally take for granted—until it goes away.

From a computer's viewpoint, an AC power line is a necessary evil. The power line may carry all sorts of messy electromagnetic interference (EMI), which is a serious problem for computers and their peripherals. Electric motors, refrigerators, and similar equipment can be devastating to a computer. There's no such thing as too much EMI filtering.

We know someone who bought a very fancy microcomputer and proudly took it home, only to be crestfallen when he saw it in action. It was utterly flaky, forever losing data and glitching. But every time he took it back to his friendly dealer, it was on its best behavior. The dealer must have been convinced that our friend didn't understand how to use an "on" switch. Eventually we went to have a look. The computer was in a family room adjacent to a blank wall. We innocently asked what was on the other side of the wall.

"Oh, just an upright food freezer," he said.

Sure enough, the freezer was within two feet of the computer, running on the same electrical circuit! Moving the computer to another room and placing it on a different circuit made it behave like the Rolls Royce of microcomputers it was supposed to be.

The voltage spikes, "brownouts," and power outages that plague commercial AC mains are also a menace to personal computers. Spike suppressors aren't very expensive; they should be included in every installation. Many radio amateurs have bought (or built) AC line voltage regulators to protect against brownouts as well as battery backup systems ("uninterruptable power supplies") to prevent data loss during a full-blown power outage. If you want to be safe (especially when using a program such as a contest logger), that step is highly recommended.

Murphy's Law and Floppy Disk Foibles

Computer owners also tend to have Murphy's Law tales about floppy disks and disk drives. There is really no substitute for disk drives if you need rapid data retrieval at a moderate price. But they are susceptible to all sorts of failures, often at the worse possible times. We won't belabor the classic advice about backing everything up, but we do have a few other suggestions.

One is that disk drives get out of alignment. If a drive gradually produces more and more disk errors—especially on disks that another drive doesn't complain about—it may be realignment time. Many radio amateurs who own good oscilloscopes do their own alignment work. All you need in addition to the scope is an analog alignment disk, usually available from the manufacturer. Ask around and you'll probably find someone in your local computer or radio club who can take you through the process, step by step. Failing that, you may have to search for a commercial shop to do the job. That may be difficult, and the going rate is typically about $50 for a 5.25-inch drive.

Disk drives have another adjustment that is much simpler: the speed adjustment. The speed potentiometer is located in different places on various brands of drives; a technical manual will tell you where to look. Most drives are equipped with a speed adjustment pattern on one pulley. If you operate the drive under fluorescent illumination, the speed can be adjusted until the pattern appears stationary. Most drives are intended to run at 200 or 300 RPM.

These days disk drives come in a bewildering array of configurations. If you find a bargain on surplus drives, we'd suggest you be absolutely certain they will work in your system before you snap them up. For example, Apple drives use a special logic board; the typical bare drive will not work in an Apple system without major modifications. And while TRS-80s use conventional drives, the Model I and Model III require different types. The Model III uses 40-track drives, and will not boot up on most 35-track drives—at least not with the Radio Shack-supplied TRSDOS operating system. And not all brands of 40-track drives will work, either.

A growing number of computers today use double-sided disk drives (i.e., drives with two read/write heads so they can access data on both sides of a floppy disk). This results in a dramatic increase in disk storage capacity at a modest increase in cost, but it further reduces the compatibility and interchangeability of drives made by various manufacturers.

In the next room we have five brand new white-elephant disk drives, monuments to our quest for bargains that turned out not to be bargains. Our advice: find out exactly what brand and type of disk drives your system is designed to use, and buy only that type if you want to add more drives. This is not to say you have to buy the drive from the mainframe manufacturer—just the same brand of drive itself. And if you don't care to heed that advice, come and see us. We have some drives for sale . . . cheap!

Computer hobbyists talk a lot about track-stepping speed in disk drives, and various speed-up schemes are making the rounds. Unfortunately, they often cause more problems than they solve. In the first place, the track-to-track stepping process is just one of several factors that determine disk access speed. Tripling the stepping speed may yield only a slight improvement in overall response time.

Besides, when you push disk drives to faster stepping speeds, you invite errors: a marginally aligned drive may work fine at a slow stepping speed, but the problem will almost certainly become more pronounced as the speed is increased. We think it's a lot wiser to be sure of accurate data transfer than to shave a few milliseconds off the access time.

Much has been written about the virtues and foibles of floppy disks themselves. We'll mention just a few of them. One is that we've found generic-label disks to be a good buy in most cases. The software and manuscript for this book were almost entirely stored on no-name disks purchased at a large discount department store. Please don't come looking for us if the particular no-name disks you buy turn out to be turkeys, but the generics are often just as good as the more expensive name-brand products.

Another suggestion: single-sided disks often work in double-sided drives. And they can be turned over so both sides can be used (one at a time) in single-sided drives. In most cases, the only difference between a single-sided floppy disk and one certified as "double sided" is that a double-sided disk has been tested (and found to be error-free) on both sides. You can format and verify the untested back side yourself. If it turns out to be good, you got a two-sided disk for the price of a single-sided one. If not, you still have one good side to use.

Finally, if you're new to microcomputers you probably wonder why there's all the fuss about disk and disk drive incompatibility. After all, any brand of phonograph record will play on any brand of turntable. A cassette tape carrying any recording label will work on most any brand of cassette player. Why should floppy disks be different?

The answer, of course, is that they shouldn't be. But they are. Although many microcomputers use the same type of "soft sectored" 5.25-inch floppy disks, the uniformity ends there. Every disk operating system places data on the disks in its own unique fashion. The only way you can make an Apple disk work in a TRS-80 or an Atari, for instance, is to reformat it—and that destroys whatever data was on the disk.

You can readily transfer text files and BASIC programs between two dissimilar computers via a telephone modem (or a hardwired interconnection between the two systems), but you probably can't swap floppy disks with a friend down the block who owns another brand of computer. In many cases you can't even interchange disks between different computer models made by the same manufacturer—at least not without going through a conversion process on each disk.

Worst of all, when you use two or more operating systems on the very same computer, disks created under one operating system usually cannot be read by the others. Apple II CP/M disks cannot be read directly

by an Apple II that is running the normal Apple Disk Operating System. Likewise, a TRS-80 disk created under one of the enhanced operating systems such as NEWDOS or DOSPLUS cannot be read by TRSDOS without special efforts. And so it goes. Rampant incompatibility is the rule in disk operating systems.

Standardization: An Impossible Dream?

Unfortunately, rampant incompatibility isn't just a problem with floppy disk operating systems—it's a nearly universal problem in the microcomputer industry.

Perhaps it's inevitable that a young and growing industry would have these kinds of problems, but some of us wonder if they shouldn't be solved now that the industry is maturing.

One thing is clear: just about everyone suffers from these incompatibility problems. Perhaps some manufacturers have felt they could enhance their profits by making it difficult for others to provide software and add-on accessories for their systems. But in the long run, it doesn't often work that way. When no one's disks can be read by anyone else's disk system software, it hurts everybody from the beleaguered dealer (who must stock many different incompatible products) to the end user (who finds his access to software needlessly limited).

And the incompatibility of the various computer languages and system commands makes matters worse. We said earlier that many computer users have adopted a philosophy of ignoring every system but their own. That's an unfortunate but understandable response when the commands are so different from system to system. What is a perfectly good command on one brand of computer is nothing but a "syntax error" on others. And as the next chapter explains, the lack of standardization causes serious problems for someone writing programs in BASIC—programs that have to work on a variety of different computers.

As owners of six different brands of computers, we find this situation maddening. We're forever typing in perfectly good Apple commands on the TRS-80, Commodore commands on Apples, and CP/M commands on all of the computers—only to be rebuffed.

Suppose every automobile manufacturer observed this kind of I'll-do-it-my-way philosophy. Someone who might be perfectly competent in a Chevrolet couldn't get behind the wheel of a Ford or Toyota without inviting a catastrophe. Not everything is standardized in the automobile industry, but the state of affairs is much better than it is in the small computer field. Even amateur radio equipment manufacturers generally use standardized labels to designate various functions. Everyone agrees about the meaning of such phrases as "RF GAIN", "BANDSWITCH", and "NOISE BLANKER." But in computers, everybody has his own commands.

Eventually, the non-standardization in microcomputers simply has to be corrected—but when? And how long will all of us be inconvenienced before that? In this book, we devote a lot of space to the problem of adapting programs from one version of BASIC to others. Even though

Applesoft BASIC, TRS-80 Disk BASIC, and the CP/M version of BASIC-80 are all adaptations of Microsoft BASIC, they all have differences—sometimes major ones that require extensive program revisions.

Many people feel that the CP/M operating system holds the key to a solution. As we indicated earlier, if a computer runs CP/M, the commands are essentially standardized. If modified for CP/M, an Apple behaves exactly like a CP/M-equipped Northstar, Xerox 820, Heath H89, Osborne I, or any of several dozen other brands. Commodore is developing a CP/M adaptation for its model 64, and CP/M will run on the TRS-80 Model II or IV (or other models, with additional modifications). This is a major step forward. But almost all computer manufacturers continue to offer—and promote—their own proprietary operating systems.

Perhaps that's inevitable, but wouldn't it be great if everyone could agree on common words for standard functions that must be performed on all computers? Why not have a universal command to call up a disk file directory? And why not a standard command to copy a file, or to initialize (or *format*, if you wish) a new disk. And when you rename a file, you put the old name first in the Apple and TRS-80 disk operating systems—but you have to put the new name first in CP/M and on the Commodore. And so it goes with command after command.

Many problems of this sort have been solved in other industries through voluntary coordination efforts, led by trade associations. Clearly, more industry-wide coordination is needed in the personal computer industry.

CHAPTER

5

BASIC: ONE LANGUAGE, MANY DIALECTS

Of all the computer languages created in the last three decades, BASIC is by far the most popular for personal computers. BASIC is an acronym for *Beginner's All-Purpose Symbolic Instruction Code.* However, BASIC long ago ceased to be a language just for beginners. Developed at Dartmouth College in the mid-1960s by John Kemeny and Thomas Kurtz, BASIC is almost always referred to today by its acronym and not by its original name.

BASIC is what is called a "high-level language," as opposed to the zeroes and ones of binary code in machine language or the mnemonic abbreviations of assembly language. A program written in assembly language executes much faster than one written in a high-level language, because high-level languages use syntax and commands that humans (but not machines) readily understand. The commands of a high-level language have to be "interpreted" for the computer by, appropriately enough, a program called an interpreter. As we said in the last chapter, a BASIC program can be "compiled" into a machine-readable form by a compiler, and that makes it run at something much closer to the speed of an assembly-language program.

But what BASIC lacks in speed, it often makes up in other ways. BASIC programs are much easier to write and revise than those written in assembly language. And an assembly language program for one particular microprocessor chip must usually be rewritten before it will work on others. Each one has its own "instruction set," consisting of mnemonic abbreviations that stand for various computer functions. In addition,

assembly-language programs that work on one computer will not ordinarily work even on other brands of computers that use the same CPU, due to differences in hardware configuration and operating systems.

A notable exception, of course, is the case of computers that share a common operating system, such as CP/M. Both assembly-language and BASIC programs written to run under CP/M can be moved from one CP/M-based computer to another with few if any revisions—except for such niceties as video display characteristics and printer interfacing.

Nevertheless, the difficulty of writing programs in assembly language—and the normal difficulty of making an assembly-language program intended for one system run on others—has led to a great reliance on high-level languages such as BASIC. Because we wanted programs that could be used on a variety of different computers, every program in this book was written in BASIC.

However, as standard as BASIC has become in the microcomputer industry, every brand of personal computer implements BASIC in a slightly different way—again with the notable exception of systems running the same version of BASIC. In fact, Microsoft BASIC, like CP/M itself, has become something of an industry standard. Happily, the version of Microsoft BASIC that runs on TRS-80 computers (TRS-80 Disk BASIC) is virtually identical to the CP/M and the IBM PC versions. As we'll explain shortly, the TRS-80 listings of our programs will run on CP/M systems and the IBM PC with only very minor changes. Although Applesoft BASIC was also adapted from Microsoft BASIC, Apple syntax differs sufficiently that we chose to include separate listings of most programs for Apple, as we did for the Commodore line.

Applesoft BASIC includes graphics features not found in the standard Microsoft BASIC, but it lacks some of the key commands that are used in handling data files and printing out well-formatted hard copies. That forced us to make major revisions in the various programs in this book as we adapted them for Apple.

And when we began converting the programs so they would run on Commodore computers, we encountered still other variations in the BASIC commands and syntax. The Commodore 64 and VIC-20 also have attractive graphics and sound capabilities. But again, the data-handling and printout commands are quite different from those on other popular microcomputers.

As much as possible, we also tried to make the programs compatible with the inexpensive Timex/Sinclair 1000 (ZX-81) computer. However, the Timex lacks provisions either for data files or for data statements within a program. That makes it quite difficult to run programs that handle large quantities of data, as many programs in this book do. The programs that primarily perform mathematical functions can be adapted to the Timex, however, and instructions for doing so are included with some of the listings. There is a separate Timex listing of the mathematically sophisticated moontracking program—which probably represents one of the most impressive things this economical little computer will do.

Although BASIC programs are transportable from one system to another (in that much of the logic and program structure remains

unchanged), the various versions of BASIC differ sufficiently that the conversion process can be time-consuming. We probably spent as much time getting the programs in this book to work under the various dialects of BASIC as we did writing them in the first place. It's a little like listening to English as spoken in England, Brooklyn, and Alabama: it may be the same language, but there are obvious differences in vocabulary and style.

In this chapter, we summarize some of those differences. It is beyond the scope of this book to present a full discussion of BASIC programming techniques or syntax. During the rest of this chapter we'll assume you know a little about BASIC. If you don't, however, you can still use these programs on most popular computers. You don't really have to understand BASIC to make good use of BASIC programs.

Our goal in this chapter is simply to illustrate some of the differences in BASIC as it is implemented on various computers, and to point out some of the quirks in various systems that may affect the way the programs run. We hope that will help you get the programs up and running with a minimum of difficulty.

The MINI-LOGGER: A Sample Program

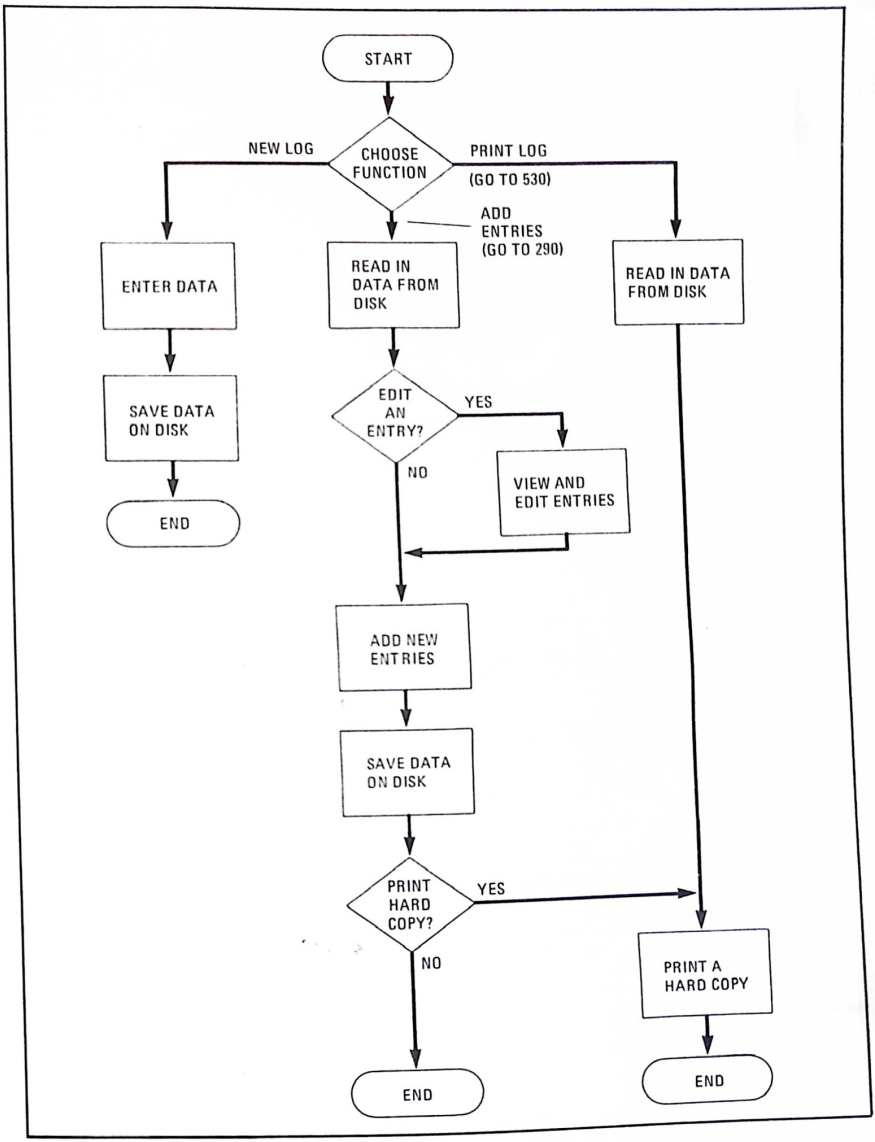
To illustrate some of the major differences in BASIC as it works with various brands of personal computers, we wrote a short program to perform routine ham radio logging functions. Actually, this program will store and retrieve any kind of data—not just logbook entries. It merely stores strings of up to 38 characters (letters and numbers) in a "data file" on a floppy disk, and retrieves the data for later editing or new additions. And the program includes a hard-copy printout function.

We chose this as a specimen program not only because it's short (we made sure the listing didn't get longer than a single page, resisting the temptation to add more features in the process), but also because it does two of the things that are handled most differently in the various adaptations of BASIC: data file handling and hard copy printouts.

As shown in the flowchart of the MINI-LOGGER program and the TRS-80 – CP/M, Apple, and Commodore listings, the user has three choices: (1) create a new data file for log entries, (2) edit the existing entries and add new ones, and (3) print out the full log. We realize a computerized logging program should do more than that, and a logger with much more complete data-processing capabilities appears in the next chapter.

Nevertheless, the MINI-LOGGER should be of interest to those looking for a quick and simple way to computerize routine logkeeping. But its main purpose is to demonstrate the differences in programming techniques for Apple, Commodore, TRS-80 and the CP/M version of Microsoft BASIC.

As the flowchart and listings indicate, if the user wants to create a new log file, the logbook entries are stored in a string array in memory and then written into a sequential data file on disk. The program uses a dimension statement, setting up an array big enough for 1000 logbook entries. Each entry may be up to 38 characters long (see line 230). The



Flowchart of the MINI-LOGGER program.

```
10 REM SIMPLIFIED LOGBOOK PROGRAM (TRS-80 AND CP/M VERSION)
20 REM (C)1983 BY HAYDEN BOOK CO.,INC.
30 PRINT CHR$(26):REM CLEAR SCREEN (CLS ON TRS-80)
40 REM CLEAR 10000 (NEEDED ON TRS-80)
50 DIM LG$(1000):J=1
60 PRINT "       SIMPLIFIED LOGBOOK PROGRAM"
70 PRINT:INPUT "WHAT IS YOUR CALL";CL$
80 PRINT:PRINT:PRINT "HERE ARE YOUR OPTIONS:"
90 PRINT:PRINT "1. CREATE A NEW LOG (DELETES OLD ONE)"
100 PRINT "2. ADD ENTRIES TO EXISTING LOG"
110 PRINT "3. PRINT HARD COPY OF LOG"
120 PRINT:INPUT "CHOICE 1, 2, OR 3";C
130 IF C>3 THEN PRINT "INVALID CHOICE-":GOTO 80
140 IF C<>1 THEN 290
150 PRINT:PRINT "CREATING A NEW LOG..."
160 PRINT:PRINT "YOU MAY NOW ENTER CONTACTS ONE AT"
170 PRINT "A TIME, USING ANY FORMAT YOU WISH,"
180 PRINT "PROVIDED EACH ENTRY HAS NO MORE THAN"
190 PRINT "38 CHARACTERS.  DON'T USE COMMAS IN"
200 PRINT "YOUR ENTRIES."
210 PRINT:PRINT "LOG ENTRY ('XX' WHEN DONE)"
220 INPUT LG$(J)
230 IF LEN(LG$(J))>38 THEN PRINT "ENTRY TOO LONG":GOTO 210
240 IF LG$(J)="XX" THEN 260
250 J=J+1:GOTO 210
260 OPEN "O",1,"LOGBOOK"
270 FOR N=1 TO J:PRINT#1,LG$(N):NEXT N
280 CLOSE:END
290 OPEN "I",2,"LOGBOOK"
300 INPUT#2,LG$(J)
310 IF LG$(J)="XX" THEN 330
320 J=J+1:GOTO 300
330 CLOSE
340 IF C=3 THEN 530
350 PRINT:INPUT "WANT TO SEE PREVIOUS ENTRIES (Y/N)";S$
360 IF S$<>"Y" THEN 440
370 PRINT:FOR N=1 TO J-1:PRINT N;" ";LG$(N):NEXT N
380 PRINT:INPUT "WANT TO CHANGE AN ENTRY (Y/N)";CH$
390 IF CH$<>"Y" THEN 440
400 INPUT "WHAT #";X
410 PRINT:PRINT "OLD ENTRY: ";LG$(X)
420 INPUT "NOW TYPE CORRECTED ENTRY";LG$(X)
430 PRINT "CORRECTED TO:  ";LG$(X)
440 PRINT:PRINT "REMINDER: NO COMMAS, 38 CHAR MAX"
450 PRINT "NEW ENTRY ('XX' WHEN DONE)"
460 INPUT LG$(J)
470 IF LG$(J)="XX" THEN 490
480 J=J+1:GOTO 450
490 OPEN "O",2,"LOGBOOK"
500 FOR N=1 TO J:PRINT#2,LG$(N):NEXT N:CLOSE
510 INPUT "DO YOU WANT TO PRINT OUT THE LOG (Y/N)";LP$
520 IF LP$<>"Y" THEN END
530 INPUT "TURN ON PRINTER. <CR> WHEN READY";CR$
540 LPRINT "          LOG BOOK OF ";CL$:LPRINT
550 FOR N=1 TO J-1:LPRINT N;" ";LG$(N):NEXT N:LPRINT
560 END
```

TRS-80 and CP/M listing of MINI-LOGGER program.

```
10  REM  SIMPLIFIED LOGBOOK PROGRAM (APPLE VERSION)
20  REM  (C)1983 BY HAYDEN BOOK CO.,INC.
30  HOME : REM  CLEAR SCREEN
40 D$ =  CHR$ (4)
50  DIM LG$(1000):J = 1
60  PRINT "        SIMPLIFIED LOGBOOK PROGRAM"
70  PRINT : INPUT "WHAT IS YOUR CALL";CL$
80  PRINT : PRINT : PRINT "HERE ARE YOUR OPTIONS:"
90  PRINT : PRINT "1. CREATE A NEW LOG (DELETES OLD ONE)"
100  PRINT "2. ADD ENTRIES TO EXISTING LOG"
110  PRINT "3. PRINT HARD COPY OF LOG"
120  PRINT : INPUT "CHOICE 1, 2, OR 3";C
130  IF C > 3 THEN  PRINT "INVALID CHOICE-": GOTO 80
140  IF C <  > 1 THEN 290
150  PRINT : PRINT "CREATING A NEW LOG..."
160  PRINT : PRINT "YOU MAY NOW ENTER CONTACTS ONE AT"
170  PRINT "A TIME, USING ANY FORMAT YOU WISH,"
180  PRINT "PROVIDED EACH ENTRY HAS NO MORE THAN"
190  PRINT "38 CHARACTERS.  DON'T USE COMMAS IN"
200  PRINT "YOUR ENTRIES."
210  PRINT : PRINT "LOG ENTRY ('XX' WHEN DONE)"
220  INPUT LG$(J)
230  IF  LEN (LG$(J)) > 38 THEN  PRINT "ENTRY TOO LONG": GOTO 210
240  IF LG$(J) = "XX" THEN 260
250 J = J + 1: GOTO 210
260  PRINT D$;"OPEN LOGBOOK": PRINT D$;"WRITE LOGBOOK"
270  FOR N = 1 TO J: PRINT LG$(N): NEXT N
280  PRINT D$;"CLOSE LOGBOOK": END
290  PRINT D$;"OPEN LOGBOOK": PRINT D$;"READ LOGBOOK"
300  INPUT LG$(J)
310  IF LG$(J) = "XX" THEN 330
320 J = J + 1: GOTO 300
330  PRINT D$;"CLOSE LOGBOOK"
340  IF C = 3 THEN 530
350  PRINT : INPUT "WANT TO SEE PREVIOUS ENTRIES (Y/N)";S$
360  IF S$ <  > "Y" THEN 440
370  PRINT : FOR N = 1 TO J - 1: PRINT N;" ";LG$(N): NEXT N
380  PRINT : INPUT "WANT TO CHANGE AN ENTRY (Y/N)";CH$
390  IF CH$ <  > "Y" THEN 440
400  INPUT "WHAT #";X
410  PRINT : PRINT "OLD ENTRY: ";LG$(X)
420  INPUT "NOW TYPE CORRECTED ENTRY";LG$(X)
430  PRINT "CORRECTED TO:  ";LG$(X)
440  PRINT : PRINT "REMINDER: NO COMMAS, 38 CHAR MAX"
450  PRINT "NEW ENTRY ('XX' WHEN DONE)"
460  INPUT LG$(J)
470  IF LG$(J) = "XX" THEN 490
480 J = J + 1: GOTO 450
490  PRINT D$;"OPEN LOGBOOK": PRINT D$;"DELETE LOGBOOK"
492  PRINT D$;"OPEN LOGBOOK": PRINT D$;"WRITE LOGBOOK"
500  FOR N = 1 TO J: PRINT LG$(N): NEXT N: PRINT D$;"CLOSE LOGBOOK"
510  INPUT "DO YOU WANT TO PRINT OUT THE LOG (Y/N)";LP$
520  IF LP$ <  > "Y" THEN  END
530  INPUT "TURN ON PRINTER. <CR> WHEN READY";CR$
532  PRINT D$;"PR#1"
540  PRINT "          LOG BOOK OF ";CL$: PRINT
550  FOR N = 1 TO J - 1: PRINT N;" ";LG$(N): NEXT N: PRINT
560  PRINT D$;"PR#0": END
```

Apple listing of MINI-LOGGER program.

```
10 REM SIMPLIFIED LOGBOOK PROGRAM (C64)
20 REM (C)1983 BY HAYDEN BOOK CO.,INC.
30 OPEN 130,2,3,CHR$(136):REM 1200BAUD, 2 STOP BITS, LF W/CR IF FILE#>127
40 PRINT CHR$(147):REM CLEAR SCREEN
50 DIM LG$(1000):J=1
60 PRINT "       SIMPLIFIED LOGBOOK PROGRAM"
70 PRINT:PRINT "WHAT IS YOUR CALL":INPUT CL$
80 PRINT:PRINT:PRINT "HERE ARE YOUR OPTIONS:"
90 PRINT:PRINT "1. CREATE A NEW LOG (DELETES OLD ONE)"
100 PRINT "2. ADD ENTRIES TO EXISTING LOG"
110 PRINT "3. PRINT HARD COPY OF LOG"
120 PRINT:INPUT "CHOICE 1, 2, OR 3";C
130 IF C>3 THEN PRINT "INVALID CHOICE-":GOTO 80
140 IF C<>1 THEN 290
150 OPEN 2,8,2,"@0:LOGBOOK,S,W"
160 PRINT:PRINT "CREATING A NEW LOG..."
170 PRINT:PRINT "YOU MAY NOW ENTER CONTACTS ONE AT"
180 PRINT "A TIME, USING ANY FORMAT YOU WISH,"
190 PRINT "PROVIDED EACH ENTRY HAS NO MORE THAN"
200 PRINT "38 CHARACTERS.  DON'T USE COMMAS IN"
210 PRINT "YOUR ENTRIES."
220 PRINT:PRINT "LOG ENTRY ('XX' WHEN DONE)"
230 INPUT LG$(J)
240 IF LEN(LG$(J))>38 THEN PRINT "ENTRY TOO LONG":GOTO 230
250 IF LG$(J)="XX" THEN 270
260 J=J+1:GOTO 230
270 FOR N=1 TO J:PRINT#2,LG$(N):NEXT N
280 CLOSE 2:CLOSE 130:END
290 OPEN 3,8,3,"LOGBOOK,S,R"
300 INPUT#3,LG$(J)
310 IF LG$(J)="XX" THEN 330
320 J=J+1:GOTO 300
330 CLOSE 3
340 IF C=3 THEN 530
350 PRINT:PRINT "WANT TO SEE PREVIOUS ENTRIES (Y/N)":INPUT S$
360 IF S$<>"Y" THEN 440
370 PRINT:FOR N=1 TO J-1:PRINT N;" ";LG$(N):NEXT N
380 PRINT:PRINT "WANT TO CHANGE AN ENTRY (Y/N)":INPUT CH$
390 IF CH$<>"Y" THEN 440
400 INPUT "WHAT #";X
410 PRINT:PRINT "OLD ENTRY: ";LG$(X)
420 PRINT "NOW TYPE CORRECTED ENTRY":INPUT LG$(X)
430 PRINT "CORRECTED TO:  ";LG$(X):PRINT
440 PRINT:PRINT "REMINDER: NO COMMAS, 38 CHAR MAX"
450 PRINT "NEW ENTRY ('XX' WHEN DONE)"
460 INPUT LG$(J)
470 IF LG$(J)="XX" THEN 490
480 J=J+1:GOTO 450
490 OPEN 2,8,2,"@0:LOGBOOK,S,W"
500 FOR N=1 TO J:PRINT#2,LG$(N):NEXT N:CLOSE 2
510 PRINT "DO YOU WANT TO PRINT OUT THE LOG (Y/N)":INPUT LP$
520 IF LP$<>"Y" THEN CLOSE 130:END
530 PRINT "TURN ON PRINTER. <CR> WHEN READY":INPUT CR$
540 CMD 130
550 PRINT "          LOG BOOK OF ";CL$:PRINT
560 FOR N=1 TO J-1:PRINT N;" ";LG$(N):NEXT N:PRINT#130
570 FOR J=1 TO 2000:PRINT;:NEXT J
580 CLOSE 130:END
```

Commodore listing of MINI-LOGGER program.

array was set to handle 1000 entries because 1000 38-character entries requires 38K of free memory, which is about all the typical 48K or 64K microcomputer has. If each entry were shorter, there would be room for more entries. And, of course, the more complex logging programs in this book store most of the data in random access disk files rather than in RAM, thus making it possible to handle longer entries—and many more of them.

The Commodore 64 listing will run properly on a VIC-20, provided the dimension statement in line 50 is revised from "DIM LG$(1000)" to "DIM LG$(30)". An unexpanded VIC-20 has only about 3500 bytes of free memory. Since the MINI-LOGGER program itself occupies nearly half of that, there is room for only about 30 logbook entries in memory at one time.

The value of 38 characters per entry was chosen merely so each entry would fit on one line of the video display on a Commodore 64 or Apple II. A much longer string would be just as practical.

Once a logbook data file is written to disk, it can be retrieved for review, editing and additions. When the user wants to make changes or additions, the entire file is loaded into RAM, and any entry can be edited. The entry to be revised is called up by its consecutive line number, which prints out both on the screen and in the hard copy. After correcting an entry, the user can enter more data—or just save the file back to disk. This, of course, involves writing over the old data file with a new one. On both the Apple II and Commodore, this requires a special procedure.

Speaking of writing over programs, the MINI-LOGGER's data file is named "Logbook," but that is also the name of the more comprehensive logging program presented in the next chapter. If you have this package of programs on disk, be sure you don't wipe out the big logger program by having it on the same disk you use for the data file that goes with this program.

After saving the log entries on disk, the MINI-LOGGER provides a hard-copy printout if the user wishes—again demonstrating a feature that requires considerably different syntax on various computers.

Comparing Disk Access Commands

If you'll note lines 260 through 330 of the MINI-LOGGER, you'll see a good example of the varying procedures for reading and writing data to sequential files on disk. In the CP/M and TRS-80 procedure, to open a file for data *input* one says:

```
290 OPEN "I",2,"LOGBOOK"
```

The "I" stands for input. The number 2 is an arbitrary file number. In BASIC-80 for CP/M, ordinarily three files may be open at once, numbered 1, 2, and 3. The files may be identified with a number sign (as in OPEN "I",#2,"LOGBOOK"), but that feature is optional and we don't use it to maintain compatibility with TRS-80 Disk BASIC.

Once you have the TRS-80 or CP/M disk file open, you can read data from it as is done in line 300:

```
300 Input#2,LG$(J)
```

After all of the data is read into memory, the file may be closed on a TRS-80 or CP/M system with a command such as "CLOSE 2" or simply "CLOSE."

If the intention is to write data out to a disk file instead of reading it in, a similar procedure is used, as shown in lines 260–290 in the TRS-80–CP/M MINI-LOGGER listing. The "O" stands for output, and the command to write is "PRINT#1" (or 2, etc.).

What happens if you want to read data in—which requires opening an input file—and write the data back out again, which requires an output file? You have two choices. You can either read the data into memory, close the file, and reopen it as an output file (as we do in this program), or you can create a separate output file for the data under a different name, delete the original file, and then rename the new file to the original name.

To see the differences between the CP/M or TRS-80 method of handling sequential data files and the Apple or Commodore methods, compare the first listing with the other two. In the Apple version, you open a file and prepare to read or write to it this way:

```
290 PRINT D$;"OPEN LOGBOOK":PRINT D$;"READ LOGBOOK"
```

If you do programming in Applesoft, you surely know and respect the "PRINT D$" command. All system commands that must be executed from a BASIC program must begin with the character, "control-D." Since that's a character that doesn't show on the screen, the customary way to handle it is to make a string variable (D$ for simplicity) equal to control-D, which is the character #4 on the ASCII table of character codes. If you check line 40 in the Apple MINI-LOGGER listing, you'll see where we've set the value of D$.

Once you've opened a file and issued the command to "READ" or "WRITE" to the file, be careful: every Applesoft input or print statement reads data from (or sends data to) that file until you use the "PRINT D$:"CLOSE . . ." routine to close the file. If you inspect the syntax in the Apple MINI-LOGGER listing, you'll see that the files are repeatedly OPENed and CLOSEd in this fashion so print commands can be directed to the screen or printer, and so input can be received from the keyboard.

There is one additional feature in Applesoft: you cannot simply write over an existing file and create a new one the way you can in the TRS-80 and CP/M versions. Instead, you have to first open the old file, delete it, and then open it anew to receive data. We jump through those programming hoops in lines 490–492 of the Apple MINI-LOGGER listing.

On the Commodore, there are still other things that must be done to use data files on disk. In the Commodore MINI-LOGGER listing, you'll see this line:

```
290 OPEN 3,8,3,"LOGBOOK,S,R"
```

That opens a sequential input file (the "S,R" stands for "sequential, read"). The first numeral 3 is the arbitrary file number, as in TRS-80 Disk BASIC. Once the file is open, you say "INPUT#3,LG$(J)" to read data from that particular file, just as you would on a TRS-80. The "8" in the OPEN statement is the standard number to identify a single Commodore disk drive. This feature works much like the "Slot 6" provision in Applesoft, in which you tell the computer where to find the disk controller card. However, the Apple will assume the disk controller is in slot 6 unless you tell it otherwise. With Commodore, you have to tell it each time to access "8" (or "9" in a typical two-drive system).

The third number (again "3") is the command channel number. In most instances that can also be an arbitrary number, but it has to be there. As indicated earlier, the "S" after the file name says it's a sequential data file and not another type, and the "R" (for "READ") says it's an input file. An output file is opened with a "W" (for "WRITE") in place of the "R."

If you'll notice line 490 in the Commodore MINI-LOGGER listing, the Commodore has another special way of handling situations in which you want to write over an existing file with new information. You don't have to go through the OPEN-DELETE-OPEN-WRITE routine as in Applesoft, but you do use a special character: "@". The designation "@0:" says the file is to go on drive zero, and that it's an existing file that should be written over.

Commodore users must exercise special care in opening and reopening sequential files for output. If an attempt is made to open a new data file for output when a file with the same name is already present on disk, the disk directory may be damaged in unpredictable ways, resulting in the loss of programs or other data files. If in doubt, include the "@" command when opening sequential data files for output.

To summarize, every system has its own way of accessing data files, and they're almost all different. And all of this is just for *sequential* data files, in which you read from the beginning to the end with no jumps to the middle, and only read from or write to a specific file—but not both at once. For *random access files*—which allow you to read or write to the same file, and to jump in anywhere along the way—the procedures are even more varied from system to system. Many of the programs in this book use random access as well as sequential data files. Unless you're an experienced programmer, you should use the file access syntax exactly as listed for your particular system.

One other thing about disk files that should be mentioned is that even the nomenclature used to designate a specific disk drive in a multi-drive system varies somewhat.

In CP/M, to read or write to a file on the second drive, you'd say "B:LOGBOOK". The first drive is called drive "A:". On a TRS-80, you'd say, "LOGBOOK:1" to put a file on the second drive, since the first drive is designated as drive ":0". On Apple, you'd say, "LOGBOOK,D2" for the second drive. And on a Commodore, it would be "0:LOGBOOK", but with the file addressed to I/O device #9 instead of #8.

There's an important point to remember about this business of designating drives: *if you're going to write data files, your computer will assume they go on your logged disk drive (usually the drive you boot up on) unless you add the syntax to the file name to put them on some other disk drive. That means any program in this book will crash if you have the disk in your first drive write-protected (or full) at a time when the program tries to write to a data file there.* And, of course, if you try to read from a data file, it has to be on the first drive unless you add language to the filenames to tell the program to look for it on another drive. There is an exception to this last point: the TRS-80 Disk Operating System knows to look on other drives if it doesn't find the file it's seeking on the first drive, but Apple, Commodore, and CP/M systems won't do that.

One final thing that differs about the use of data files on various computers is the method of error-checking to see if all of the data has been read in. We simply avoided the problem by making sure the last data line in the MINI-LOGGER's data file consists of "XX," and then by writing the program so the "XX" doesn't print out. By doing that, we were able to omit the end-of-data error-trapping routine entirely, avoiding another thing that would have to be converted to make it work on various systems. Most of the sequential data files in this book do the same thing, and for the same reason. That's why the large latitude-longitude data file in Chapter 7 ends with this entry: "ZZZ","ZZZ",0,0.

Hard Copy Printout Commands

Aside from the methods of disk file access, the hard-copy printing commands probably differ the most of any commands we use in the programs in this book.

In the TRS-80 and CP/M versions of Microsoft BASIC, there's a very simple way to request a printout: "LPRINT". When you want a statement or the value of a variable to print on your display screen, you say, "PRINT", just as you would on almost any computer. To route the output to the hard copy printer (or "line printer") instead, you just substitute LPRINT. All the other syntax is exactly the same. In the TRS-80 – CP/M MINI-LOGGER listing lines 540 and 550 of the MINI-LOGGER show this command in use. Compared to other methods, it's a very easy way to print a hard copy.

Although we didn't use it in the MINI-LOGGER program, another very useful feature of the TRS-80 and CP/M versions of Microsoft BASIC is a command called PRINT USING or LPRINT USING. You'll see it in various programs in later chapters. This command allows you to set up a format for the printout by giving the computer a template for what you want. You can specify the number of decimal places, the tabular arrangement, whether or not you want commas in long numerals, the length of string variables, and other useful things. Among the best features of PRINT USING are the automatic rounding of numbers and the automatic justification of tabular columns with the decimal point always falling in the same place regardless of the length of the particular number.

Without the PRINT USING or LPRINT USING command, tabular print formatting can still be done (as it must be for many of the programs in this book), but it takes a lot more programming time and effort to do it. The availability of these commands is one of the best features of Microsoft BASIC-80. We've known business-oriented programmers who have bought the Z80 Softcard for their Apples just to get the LPRINT USING feature of Microsoft BASIC (which comes with the Softcard).

Returning to the printout feature of the MINI-LOGGER, refer to lines 532–560 of the Apple listing for the commands used on a standard Apple to get hard copy. To route output from the screen to the printer (or another peripheral device), you use a command like PRINT D$;"PR#1". If your printer interface is in some accessory slot other than #1, of course, you would use the appropriate number in place of "#1." Then the "PRINT" command sends all output to the printer rather than the screen, although with some Apple printer interface cards the printout echoes back to the screen as well. This screen echo causes Apple programmers some special problems that we'll discuss later.

After you've sent whatever you wish to the printer with the PRINT command, you have to re-route output back to the Apple's video display with the PRINT D$;"PR#0" command. However, if you are using an 80-column video board in your Apple, this has to be modified—depending on which 80-column board you have. Even if you have an 80-column board, of course, you can run your Apple II or IIe in the 40-column mode and use the standard 40-column commands. This mini-logger program, like the others in this book, was written to run on any Apple II with Applesoft BASIC, from the earliest ones to the IIe.

On a Commodore 64 or VIC-20, the procedure for sending output to a printer requires still another step. Referring to the Commodore MINI-LOGGER listing, note line 40. Before you can print, you have to open a file and a channel to the printer. Our program shows the command, OPEN 130,2,2,CHR$(136) because we use an RS-232 serial interface for printing. With the parallel Commodore interface, the second number would be 4 instead of 2.

We used "130" as a file number because the Commodore only sends a line feed with each carriage return if a number above 127 is specified. If we didn't do it, we'd have the lines all overprinting on top of each other, since the other computers we use are set up to send line feeds with carriage returns, and the automatic line feed feature is turned off in our printers. CHR$(136) sets the parameters for the RS-232 port. In our case, we wanted 1200 baud output with two stop bits. The Commodore RS-232 manual indicates which codes are required for other data rates and configurations.

Once a channel to the printer is opened on the Commodore, there are two ways to direct output there. One is the "CMD 130" command in line 532 of the Commodore program. That command is similar to Apple's "PR#1" in that once it is issued, all print commands send data to that peripheral until the command is canceled. Thus, after "CMD 130," ordinary print commands send output to the line printer. To cancel this command,

one must say "PRINT#130", which sends a blank line to the hard-copy printer as well as returning output to the screen.

Besides the "CMD" method of routing output to a hard-copy printer or other peripheral device, there's another way to do it on a Commodore, and it's preferable if you want to repeatedly transfer output from the printer to the screen and back again. You use "PRINT#" as a print command. In the MINI-LOGGER, output could be routed to the printer by saying, PRINT#130,N;" ";LG$(N) at line 550. The PRINT# command is almost as nice as saying "LPRINT" on a TRS-80. But there is one slight catch: the TAB command doesn't work at all with PRINT#, and even the SPC (space) command can't be used at the start of a print statement, although it can be used further into the statement. But there are ways around this problem, too, such as putting in leading spaces with syntax such as this:

```
PRINT#130, "    "A$;SPC(5)B$ (etc.)
```

Another problem with hard-copy printouts on the Commodore is the need to close the channel to the printer with the command "CLOSE 130". However, that command can't be used until all characters have been sent from the buffer to the RS-232 line. If the channel is closed too soon, part of the output will be cut off. The Commodore manuals recommend a method of checking to see if all characters have been sent, but we found the time delay in line 555 to be more reliable. The time delay consists of a for-loop that tells the computer to "print" 2000 times, but without printing anything in particular or even sending carriage returns anywhere. It merely keeps the computer busy for a couple of seconds so the last characters can safely escape the buffer and be on their way before the channel is closed.

One of the better features of BASIC is that it's flexible. There are usually several ways to accomplish a programming goal. If one way won't work because of the unique command structure used on a particular computer, there is often an alternative way to do the same thing that *will* work. Innovation and experimentation sometimes pay off, especially when the goal is to adapt a program that works perfectly well on one brand of computer so it will work on another.

Specific Programming Notes for Various Computers

As we adapted the programs in this book to run on various computers, we came across a number of practical problems that are hardware-specific. In this section, we'll summarize some of the major ones.

Special Considerations for TRS-80—In adapting the programs for the TRS-80, we had remarkably few problems. However, there are several things you must do to make any program work on a TRS-80. For one, because the TRS-80 does not dynamically allocate memory space for string variables, you must be sure to use a "CLEAR" command at the beginning of each program that uses many strings or string arrays. If you omit the

CLEAR statement, only a minimum of memory is allocated for string variables, and most of the programs in this book will crash as a result. Some of the programs require a great deal of string space: the DX DISPLAY and GRAYLINE programs in Chapter 7 include commands like "CLEAR 20000", setting aside 20K of RAM for string arrays. Other programs, of course, require much less string space.

Another caution: be sure the CLEAR command *precedes* the lines establishing values for any variables, because the CLEAR command wipes out all existing variables as it frees the specified amount of memory for string storage.

The CLEAR statement is not needed and should not be used in the CP/M and IBM PC versions of the programs, since Microsoft BASIC (versions 5.0 and later) dynamically allocates memory for string variables.

Another thing to remember about the TRS-80 is that you must specify the number of files you want space set aside for when you call up BASIC. And you have to say so if you want variable-length records for these files. All of the logging programs and the DUPECHECKER use variable length random access files, so the "3V" option must be specified when entering TRS-80 Disk BASIC to run these programs. The TRS-80 manual explains this feature in more detail.

One additional point: in the TRS-80 there is an unusual feature inherent in the functioning of the <ENTER> key (the TRS-80 equivalent of RETURN). If you respond to an INPUT statement by typing <ENTER> instead of a new value, the previous value of a string variable is retained as if it had been typed again. Many of the programs use the <ENTER> command as a quick way to skip over a question when you don't want to specifically tell the computer to do something else. In the contest logging programs, for example, when editing a previous log entry you type a RETURN when each item is displayed if you want to leave it as is, entering new data only if you wish to make a change.

To avoid incorrect editing on the TRS-80, we had to precede each input statement with another statement setting the value of the variable to a null string (e.g., A$=""). You'll see these statements throughout the TRS-80 listings. They are necessary; don't eliminate them. For that matter, they don't hurt anything (other than occupying space) in the CP/M versions of the programs. You may or may not want to delete them if you're typing a program into a CP/M system, working from the combined TRS-80 – CP/M version of the listing.

Special Considerations for Commodore—When the Commodore VIC-20 hit the market, there was considerable excitement. Here was an inexpensive computer that had a real keyboard, came with a rather complete version of BASIC, and was expandable enough that more memory and even disk drives and a printer could be added at a modest cost. And the VIC-20's big brother, the Commodore 64, offers 64K of memory for only a few hundred dollars. If you add a disk drive and printer to either one of these low-cost computers—and expand the memory in the case of a VIC-20—you can run most of the programs in this book with a minimum investment in hardware.

Five of the programs (the GRID LOCATOR, the DX CHECKER, the ANTENNA MATCHING EVALUATOR, the SKYLOCATOR, and, of course, the MINI-LOGGER) will run on an unexpanded VIC-20 if they are modified as described here and in the specific program descriptions. With enough additional memory and a Commodore 1541 disk drive, most of the others will also run on a VIC-20. However, even with the maximum allowable memory expansion, certain programs in this book will overtax a VIC-20's memory. The contest logging programs, for example, will occupy so much of the free memory even with the maximum expansion that there will be little space left to store call signs and other variables.

One unique feature of the VIC-20 that requires special attention is its 22-character-wide video display. The Commodore listings in this book were written for the Commodore 64's 40-column display. They will not look right on a 22-column display unless rearranged. For example, many of the statements will need to be divided and placed on two lines. Even then, there's only so much you can put on a screen when each line is limited to 22 characters. In view of the limited memory and 22-character display of the VIC-20, the Commodore 64 is a better choice than even a fully expanded VIC-20 for the more elaborate programs in this book. Given the extent to which the Commodore 64's price has plummeted, it offers a lot of performance per dollar—probably more than the VIC-20.

Another factor to consider in running the more complex programs on either Commodore computer is disk access speed. The Commodore disk operating system seems to be considerably slower than the others we've tested when it must store and retrieve large quantities of data. The slow disk access may test some people's patience; it does ours, especially when we're running the contest logging programs. Fortunately, rapid disk access isn't crucial to many of the programs. Even the DUPECHECKER program, which stores all vital data in RAM until there is a SAVE command, is unaffected by this problem. A Commodore 64 will do contest dupechecking (or anything else that doesn't involve recurring disk access) virtually as fast as an Apple II.

Commodore users should note certain other things about program syntax. For one, the Commodore program listings may need to be customized for a particular printer. While the CP/M and TRS-80 "LPRINT" command will access any printer that is properly interfaced to the computer, the Commodore hard-copy syntax varies with the type of printer. Our Commodore is set up to drive a printer via the serial RS-232 port, which is not the only way to drive a printer with a Commodore computer. The RS-232 port is initialized with a command such as "OPEN 130,2,2,CHR$(136)", as indicated earlier. If you have anything but a 1200-baud serial printer, every program will have to be revised for your specific hardware. For example, VIC printer owners should use the printer channel syntax described in that unit's manual. If you change the number "130" to something else, be sure to make that change each time the printer is addressed throughout the program. Every "PRINT#130" or "CMD130" statement must be changed to the new number.

There are several other things about Commodore BASIC programming that are unusual. For example, the maximum allowable

length of a program line is 80 characters, not the 255 characters permitted on most other systems. If you go over 80 characters, the line will crash.

Another trait of the Commodore that caused us no little confusion is that an INPUT statement must all fit on one line of the video display. Many of the programs use "prompts" with INPUT statements (i.e., instructions about what to type that appear with the INPUT command itself). Any INPUT statement that is longer than a single displayed line is likely to work incorrectly, with unpredictable (and usually bad) results. If you compare the INPUT statements in the Commodore Logger listings with those in the Apple and TRS-80 listings, you will see how we dealt with this problem.

The problem with INPUT statements that run longer than a single displayed line crops up even more often with the VIC-20's 22-column display. As indicated, the Commodore listings were written for the Commodore 64's 40-column video display. To be safe, VIC-20 users should revise all INPUT statements that include prompting messages inside quotes so that the message follows a PRINT instruction, with a separate INPUT command. Here is an example of how a line in a Commodore 64 listing can be changed for the VIC-20.

```
100 INPUT "ENTER YOUR LATITUDE IN DEGREES";LA          (original)

100 PRINT "ENTER YOUR LATITUDE IN DEGREES":INPUT LA    (VIC-20)
```

One more Commodore trait that should be reiterated here is the problem of writing over existing files, a problem we cited earlier. If a file already exists on disk and you try to "create" it as a new file for sequential output (i.e., "write") file, you may destroy crucial parts of the disk directory.

Some of these syntax requirements are more cumbersome than their equivalents on most other computers. But the Commodore line does offer good programming capabilities for the price. Even if you have a VIC-20 without a disk drive, you can run some of the programs. Commodore provides a way to store and save sequential data files on cassette, which means programs based on the worldwide latitude-longitude data base (see Chapter 7), for instance, will run if you have sufficient free memory (about 20K). As a matter of fact, you can also run those programs on a cassette-based system by converting the entire data base into a long series of DATA statements at the end of the program and use READ statements to load the data into memory. We show how to do this in Chapter 7.

Special Considerations for Apple—For Apple users, there are a number of things that may affect the way these programs execute. Because the Apple is so expandable, hundreds of different accessories are readily available and many of them affect printer interfacing or the video display features. Unfortunately, this complicates Apple programming.

All of the programs in this book will run on a standard Apple II or IIe with the standard 40-column video display. Originally, many of them had provisions for an 80-column display option. However, when the Apple IIe

was introduced, it featured a made-by-Apple 80-column board that used somewhat different commands than the older 80-column boards made by other manufacturers. Thus, what works with one 80-column board will not necessarily work properly with others. To avoid needless confusion, we revised all of the Apple programs to run in the standard 40-column format only. Experienced Apple programmers may wish to revise the Apple listings to take advantage of the 80-column features of their particular systems.

Another Apple hardware configuration problem involves printer interfaces. With some of the Apple printer interfaces we've tried, it takes a special command in the program to get 80-column hard copy. The command in most cases is control-I,"80N". However, this doesn't work with every printer interface—and it isn't needed on most of them. Even worse, the "80N" prints on the hard copy with some interfaces, messing up tabular arrangements that it took a lot of trouble to line up. There was no way to write the programs so that they would produce 80-column printouts with all printer interfaces without introducing potential glitches with some combinations of hardware.

As listed in this book, the Apple programs print properly and display properly on a standard Apple II or IIe with most parallel (and many serial) printer interfaces. Only if you have unusual hardware are modifications likely to be needed.

We should mention one other display/printout problem with some Apple printer interface cards. Some cards echo each printed character back to the video screen, while others do not. If yours does, you'll get a badly aligned version of your 80-column hard copy on the screen during printouts. With some printer interfaces (especially serial ones), nothing like that happens, and you may not see anything on your screen during a hard-copy printout. In some cases, the control-I,"80N" command suppresses this screen echo, but on other boards it doesn't help. This extra action on the screen really doesn't hurt anything, in any case. If you're not having a hard copy printed out, it doesn't happen. And if you are getting a hard copy, what matters is that the hard copy should look right. Even with the fanciest word processors, the hard-copy printout echoes on the screen with some Apple printer interfacing arrangements.

Another related problem is that the TAB function doesn't work correctly with some Apple interfaces. As a result, the printed columns may be badly misaligned. We went to considerable lengths to write printing routines that do not use the TAB command (except at the beginning of a line). We think this solves the problem for every interface, but we wouldn't be surprised if someone somewhere were to have a combination of hardware on which the printout doesn't look right.

In connection with all of these problems, we tried to write the programs so that they would work correctly on Apple computers using the most typical configurations. If your video or printed output isn't tabulated correctly, or if you get garbage on your screen at a time when it's supposed to have been cleared by the HOME function, the program doesn't suit your particular hardware.

On the bright side, most Apple owners will not see these problems, and in any case they affect only the appearance of the printout or display,

not the way the programs run. This kind of confusion is the price an Apple owner has to pay for having a computer that is so easy to expand and upgrade. None of these printout or display problems exists with most other computers—but only because the user has so little freedom to alter the hardware configuration without performing major surgery.

However, one glitch that may well affect the running of the programs on other brands of computers, and not just the Apple, arises when there is a command to send data to the printer at a time when the printer isn't turned on, isn't ready to print, or doesn't handshake with the computer correctly (see Chapter 3). Many computers will "hang" and refuse to do anything if they see a command to print at a time when they can't print. When debugging any program, don't overlook this possibility.

Special Considerations for CP/M Systems—Of all the different computers on which the programs in this book were designed to run, the CP/M systems will be the most varied in their hardware and software configurations.

If you're running CP/M 2.0 and have Microsoft BASIC-80 ("MBASIC.COM", version 5 or later), these programs should work as advertised. However, MBASIC.COM is a 24K file all by itself, so you'll need to be running a 48K CP/M system at the very minimum (and preferably 56K or 60K) in order to have enough memory to run some of the larger programs. In fact, even with a 56K CP/M system, you only have about 26K of free memory when MBASIC is loaded (remember that CP/M itself occupies some of that 56K).

Apple owners who have the Z80 Softcard and Microsoft BASIC should still consider running the larger programs in this book (such as the contest loggers) with Applesoft rather than Microsoft BASIC, since an Apple II has about 10K more free memory when running Applesoft. Applesoft will allow you to handle far more QSOs with any of the contest logging programs.

All of the programs in this book will run nicely under MBASIC on a Xerox 820, Kaypro and Osborne I, among other CP/M systems.

The major customization that will be required to make these programs run correctly on most CP/M computers with BASIC-80 has to do with the video formatting and display features. CP/M computers use a variety of different commands to perform such functions as clearing the screen or switching to inverse video.

When we wrote the CP/M versions of the programs, we used CHR$(26) (control-Z) to send a clear-screen command. This works on Televideo and some other brands of terminals, but it is by no means universal. However, we are reproducing the TRS-80 versions of the programs rather than the ones written specifically for MBASIC 5.0. Thus, the clear screen command for TRS-80 (CLS) is used. If you're running a CP/M system with MBASIC, each CLS command should be replaced with "PRINT CHR$(n)" where *n* is your clear-screen code. The text indicates the line numbers where this change is needed in each program.

Also, near the beginning of the largest programs, you'll find a string variable called "HI$" and another called "LO$". You should set these equal to your particular screen highlighting and normalizing commands. On our

terminals, CHR$(21) turns on the video inverse and CHR$(20) restores normal operation. Replace these values with your own, lest you get unpredictable results on the screen.

If your system doesn't have video highlighting, or if you decide you don't want video highlighting, you could go through each program and delete all of the PRINT HI$ and PRINT LO$ commands. A simpler solution, however, is to set HI$ and LO$ equal to what is called a "null string"—a pair of quotation marks with nothing inside them. In each program, replace the values for HI$ and LO$ with this syntax:

```
HI$="":LO$=""
```

If you do this, the program will run correctly, but without video highlighting.

Because we are reproducing the TRS-80 versions of many programs rather than the CP/M versions, many of the programs have 64-column video formats. You may want to use more tabbing to center the display on your 80-column screen. In the same vein, we have inserted INPUT statements at strategic points in the TRS-80 programs to prevent the video displays from scrolling off of the 16-line screen. If you're running CP/M, you almost certainly have a 24- or 25-line display, and you may find the frequent "<ENTER> to continue" commands distracting. You will probably prefer to eliminate them, or at least move them so they occur only at 24-line intervals.

Perhaps we should conclude the discussion of CP/M by pointing out that we made no attempt to make the programs run on any CP/M-compatible version of BASIC except Microsoft BASIC-80. MBASIC.COM is so widely accepted—and so versatile—that we felt it was clearly the appropriate choice for these programs.

Special Considerations for the IBM PC—After the programs in this book were running correctly on a variety of other computers, they were adapted for the IBM Personal Computer (PC).

We asked a friend to lend us his IBM PC for this program debugging task, which is a little like asking to borrow someone's Mercedes Benz for a hidden transmitter hunt. In either instance, you'll find out quickly just how much of a friend your "friend" is. Our friend passed the test, and we quickly learned why IBM is winning a large share of the microcomputer market: the PC is powerful, versatile, and easy to use. As it turned out, the IBM PC conversion process was the easiest of any we attempted.

The IBM PC's MS-BASIC is a very good adaptation of the widely used Microsoft BASIC. The CP/M versions of the programs will run under "BASICA" (the most powerful of three levels of BASIC on the IBM PC) with virtually no modifications. BASICA has some new commands that are not found in standard Microsoft BASIC revision 5.0, but it recognizes all of the standard MBASIC 5.0 commands.

All of the programs in this book will run correctly on an IBM PC if you follow these steps:

1. Use the TRS-80 and CP/M listing, and make all of the modifications required for CP/M *with the exception of deleting CLS*

commands. The IBM PC recognizes CLS as a clear-screen command, as does the TRS-80, so each CLS should be left as is. But be sure to delete the CLEAR statement near the beginning of each program. That command should not ordinarily be used in the IBM PC versions of these programs.

2. For the simplest possible conversion, do not add the "PRINT HI$" and "PRINT LO$" commands as you would for CP/M computers, since the IBM PC uses its own command structure to activate the video highlighting features. In listings that already have these commands in place, deactivate them by making HI$ and LO$ equal to null strings as explained earlier in this chapter.

3. At this point, each program should run correctly under "BASICA" on the IBM PC. However, if you want to include IBM-style video highlighting features, a good command for video inverse is "COLOR 0,7" and the comparable command to return to normal video is "COLOR 7,0". Insert the appropriate COLOR command wherever the CP/M version has a "PRINT HI$" or "PRINT LO$" command.

As we suggested for CP/M computers, IBM PC owners may want to revise the tabbing so the displays are centered on an 80-column line. The TRS-80 versions are set up for a 16-line, 64-column display. The tabbing modifications described in the section on CP/M are equally applicable to the IBM PC.

It should be emphasized that all of these display features are purely cosmetic; the programs will run as they should without any video display enhancements. If you have an IBM PC, we recommend that you get each program running in its "vanilla" form first, and then add whatever features you want to dress up the video display.

4. The IBM PC's internal clock, like the one in the TRS-80 Model III, can be used for the time-keeping function in the contest logging programs. The syntax differs slightly from that used on the TRS-80 in that the TRS-80 TIME$ function includes both the time and date. On the IBM, the DATE$ function returns the date, while TIME$ is the time only. This requires changing one line in each contest logging program: line 1162 in the GENERAL PURPOSE CONTEST LOGGER and line 1182 in both the Field Day and Sweepstakes versions. The IBM PC date and time-input line should read as follows:

```
1162 (or 1182) DTE$=DATE$:UTC$=LEFT$(TIME$,2)+MID$(TIME$,4,2)
```

If you wish to reset the IBM PC's clock, the method described in Chapter 8 (for the TRS-80 Model III) is generally applicable. However, the TRS-80 POKE commands described there are not needed to reset the PC clock. You can INPUT the value of TIME$ directly, using the same time format as at the initial sign-on in PC DOS. Thus, line 4220 in the time-resetting subroutine (see page XX) is not needed, and line 4210 should directly input a new value of TIME$ (in hours, minutes and seconds, separated by colons) in place of the separate values for hours and minutes (HH and MM).

But aside from the internal clock and video display features, the CP/M versions of the programs will run without modification on an IBM PC. Thanks to the wide (and growing) popularity of various versions of Microsoft BASIC, programs written for one brand of computer are becoming less difficult to adapt for others.

BASIC Peculiarities: A Summary

Given all of the ways in which both the hardware and the BASIC command structure vary from computer to computer, we felt a table summarizing the differences in major BASIC functions would be in order.

Basic Dialects: Some Major Differences

Function	CP/M (MBASIC)	TRS-80	APPLESOFT	COMMODORE
Exponents	^	[	^	^
String limits in PRINT USING	\	%	n.a.	n.a.
Clear screen	varies	CLS	HOME	PRINT CHR$(147)
Disk drive designations	A: B: etc.	:0 :1 etc.	,D1 etc.	0: 1: etc.
File names	FILENAME.BAS (8 CHAR+.BAS)	FILENAME/BAS (8 CHAR /BAS OPTIONAL)	30 CHAR SPACES OK	15 CHAR SPACES OK
Hard Copy	LPRINT	LPRINT	^D;"PR#1" PRINT ^D;"PR#0"	OPEN X,4,X, CMD X PRINT PRINT#X CLOSE X
Sequential disk input files	OPEN "I", X,"FILENAME" INPUT#X,A,B$ CLOSE X	OPEN "I",X, "FILENAME" INPUT#X,A,B$ CLOSE X	^D;"OPEN FILENAME" ^D;"READ FILENAME" INPUT A,B$ ^D;"CLOSE FILENAME"	OPEN X,8,X, "0:FILE,S,R" INPUT#X,A,B$ CLOSE X
Sequential disk output files	OPEN "O", X,"FILENAME" PRINT#X,A,B$ CLOSE X	OPEN "O",X, "FILENAME" PRINT#X,A,B$ CLOSE X	^D;"OPEN FILENAME" ^D;"WRITE FILENAME" PRINT A,B$ ^D;"CLOSE FILENAME"	OPEN X,8,X "0:FILE,S,W" PRINT#X,A,B$ CLOSE X
To set aside RAM for strings	not needed	CLEAR XXX	not needed	not needed

1. Commodore uses device #4 for printers, but #2 for RS-232
2. For random access files, see reference manuals
3. In file access, CP/M MBASIC allows # before file number
4. PRINT USING is not available on Apple, Commodore
5. See text re Apple video and printer card interfacing
6. The symbol "^D" in the Apple syntax means control-D.

The preceding table presents this summary. Of necessity, we had to leave out many details, including some that are discussed in this chapter. But we felt a quick reference guide would be useful nonetheless.

As we conclude, we should also acknowledge that in emphasizing the differences and not the similarities among these different BASIC dialects we may have inadvertently done a disservice to BASIC. BASIC is a very versatile computing tool. It is far more universally used and understood in the microcomputer world than any other language. The fact is that most BASIC commands and functions work in exactly the same way on computers ranging all the way from giant mainframes to the little Sinclair. That's why we chose to write all of the programs in BASIC in the first place.

CHAPTER

6

DATA BASE MANAGEMENT PROGRAMS FOR AMATEUR RADIO

One of the tasks at which computers excel is data base management. Computers can quickly store and retrieve large quantities of data, sorting the data in a variety of ways.

The two major programs in this chapter take advantage of that capability, applying data-base management techniques to the information-handling needs of radio amateurs.

The chapter includes a general purpose logging program called "LOGBOOK," and also a program called "THE RADIO AWARDS DATA BASE" that can be used to keep a band-by-band record of states, countries, zones, grid locators, or any other geographic unit that is the basis for an operating award. In addition, there is a short program to convert latitude and longitude to grid locators—the geographic unit on which the VHF-UHF Century Club awards program is based.

The logging program is the big brother of the MINI-LOGGER presented in the last chapter. It will store information about thousands of contacts in random-access files on floppy disk. The information may be retrieved in a variety of ways. It is possible, of course, to print out the full log. But it is also possible to have your computer sort and print out just certain contacts. For example, it will pull out QSOs with a particular station (ever wonder how many times you really have worked your friend Charley??), or with stations having a certain prefix. Or it will sort QSOs by date, band or mode. It will even flag and display a "deadbeat list" of QSOs for which no QSL has been received!

Of course, the LOGBOOK program allows full editing of every entry; you can make changes and corrections whenever you wish. And you can

go in later and add new information, such as a home address or phone number, as you learn it.

You can keep multiple logbooks as well, maintaining data files on a separate floppy disk for each one.

The RADIO AWARDS DATA BASE program is just as flexible in its sorting capabilities, but it is tailored to the needs of amateurs seeking various radio awards. This program will work with any awards program because you set up the specifications in a special file that tells the computer the essentials of how the particular awards program operates.

As with the LOGBOOK program, the AWARDS program will maintain multiple data files on separate disks. That means you can keep track of your progress toward five-band DXCC on one disk, five-band WAS (Worked All States) on another, and still another award on a third data disk.

Like the LOGBOOK, the AWARDS program includes full editing capabilities, along with the option of retrieving and displaying data in various ways. You can check the number of states (or countries or grid locators or whatever) overall or band by band. You can also call up the full details about any one geographic unit without listing all the others.

One of the program's most important features is its ability to keep track of QSL information. It will tell you how many states, countries or whatever you've worked, and how many you have confirmed. Again, you have full editing capabilities so you can modify any entry in the data files. You can call up information in various ways, and display or print the results.

You need at least one disk drive and a printer to derive full benefit from either of the data base management programs. However, this chapter also includes a short program that will run on any personal computer, including the least elaborate ones: it's called the GRID LOCATOR.

The underlying purpose of the grid locator system is to provide short and unique descriptors for any point on the globe for VHF contest and award purposes. The entire world is divided into rectangles one degree in latitude high and two degrees in longitude wide (actually, these geographic units are more like trapezoids than rectangles, since the longitude lines are not parallel but instead get closer together as you move toward the poles).

The GRID LOCATOR program has the logic of this classification system reduced to mathematical formulas. You enter the latitude and longitude for any point (except, of course, the North and South Poles, at which 18 of these grid blocks converge), and the program will display the correct locator.

THE COMPREHENSIVE LOGBOOK PROGRAM

What it does

The LOGBOOK program will store information about thousands of contacts in random access files on a floppy disk. The information may be edited and revised—and new contacts may be added—at any time. The

information may be retrieved and printed out as a whole, or the data base may be searched for specific information in the following categories:

Recent contacts;

Contacts with a particular station over a period of time;

Contacts with all stations having a certain prefix;

All contacts on a certain date or group of dates;

All contacts on a certain band or group of bands;

All contacts on a certain mode;

All contacts for which no QSL card has yet been received.

The number of QSOs that may be stored depends on floppy disk capacity. On an Apple II, a single disk will store about 2000 QSOs, including space for a 25-character comment after the QSO information.

Setup instructions

The LOGBOOK program is menu oriented. The operating procedures are explained in the menus. However, when you set up the program for the first time, there are some decisions you must make. For one thing, you have to decide on the printout format.

The program will print a log in either of two formats: with all information about each QSO on a single long line, or with a wraparound to a second line. For clarity, it is probably better to have everything on one line (see the examples of the printout, below).

If your printer will only print 80 or 85 characters per line, that severely limits the length of the comments you can include with each QSO if you want everything on just one line. The basic information about each QSO requires 67 characters, which means the "comment field" can be no more than 13 characters if everything has to fit on an 80-space line. The length of the comment field is set at line 150. If you want each QSO to fit on one 80-column line, the value of C1 in line 150 must be reduced from 25 to 13. With an 85-space line, the comment field can be 18 characters, etc.

In addition, the print width in line 30 should be changed if you want more than 70 columns per line. If you have a printer capable of operating in a condensed-type mode, or if your printer accommodates sheets of paper wider than 8½ × 11", you can include a larger comment field and still get each QSO on one line.

To adjust the size of the comment field, the only thing you have to change in the Apple version is the value of C1; that automatically determines the record size in the disk files.

However, if you change the value of C1 in the TRS-80 and CP/M version, line 202 also has to be changed. That line sets up the size of each field in the random access data files, and CM$ must be changed from 25 to whatever new length you select for your comment field if you're using either Microsoft BASIC or TRS-80 Disk BASIC.

In any version of this program, the value of C1 determines whether each log entry prints on a single line or wraps around onto a second line.

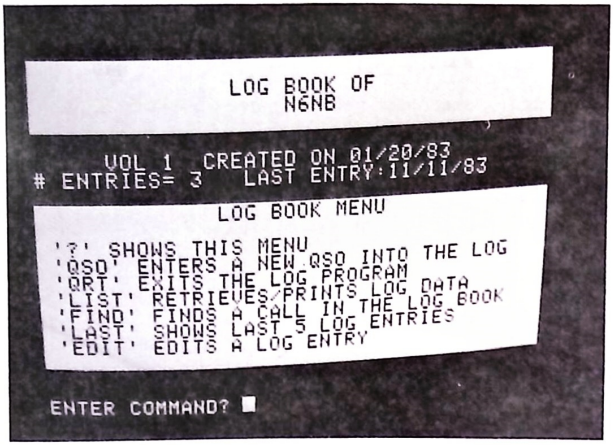
Sample of 80-column log printout.

```
                              LOG OF KC6A                          PAGE 1
                                VOL 1

                                                          HIS MY       QSL
RCD#   DATE     TIME    CALL       FREQ     MODE PWR    RST RST  QRT  S/R
---- -------- ----- ---------- -------- --- ----- --- --- ---- ---
1    01/15/83 0100 N6NB        222        FM   10              0130  /
     COMMENTS: WAYNE - TRYING NEW RIG

2    01/16/83 0200 AA6DD       145.55     FM   15    59  59  0230  /
     COMMENTS: WILL- FULLERTON

3    01/16/83 1220 K6LPF       145.55     FM   15    59  59  1400 N/
     COMMENTS: BOB FULLERTON 714 8705432

4    01/17/83 0200 N6NB        222.000    FM   15              0000  /
     COMMENTS: WAYNE

5    01/18/83 0230 N6NB        144        SSB  10              0300  /
     COMMENTS: SITE TEST FOR 10 GHZ

6    01/18/83 1410 K6LPF       144.55     FM                         /
     COMMENTS: BOB

7    01/20/83 0730 K6IBY       220.1      SSB  1300  59  59  0800 N/
     COMMENTS: JOE- COSTA MESA

8    01/20/83 0730 K6JKQ       220.1      SSB  1300  55  56  0800 2/Y
     COMMENTS: DAVE- STOCKTON

9    01/20/83 0730 WA6KOU      220.1      SSB  1300  57  59        /
     COMMENTS: BILL

10   01/20/83 1900 K6MYC       1296.025   CW   220   539 549 1920 Y/
     COMMENTS: MIKE-MORGAN HILL. WOWEE

11   01/20/83 2000 AA6S        1296.025   SSB  220   57  57  2015 Y/
     COMMENTS: BILL- VISALIA

12   01/22/83 0300 W6ABN       432.060    SSB  60    59  59  0400  /
     COMMENTS: STAN- APPLE VALLEY

13   01/22/83 0300 W6YVO       432.060    SSB  60    59  59  0400  /
     COMMENTS: DICK- ONTARIO

14   01/22/83 0300 KD6R        432.062    A3J  15    589 599 0320 N/Y
     COMMENTS: GENE- MT. PALOMAR

15   02/09/83 0100 KL7AIZ      144        EME  1000  0   0   0230 Y/
     COMMENTS: YIPPPPPEEEEEEEE!!!!

16   01/30/83      N6NB/7      144.2      SSB  160   52  51        Y/
     COMMENTS: WAYNE- UTAH PASS
```

If the length of the comment field plus 67 (the length of the standard information) exceeds the print width, the program will put each entry on two lines.

Remember that the LOGBOOK program uses a lot of disk space; do NOT attempt to run the program on a disk that is cluttered with other programs. It is probably best to put only the working copy of the LOGBOOK program on a special disk so there is as much space as possible for the data files. Another solution is to specify a second disk drive for the data files.

As the program is written, the data files go on the main drive. To change that, modify the filenames in lines 180 and 190 to include a drive


LOG BOOK OF
N6NB
VOL 1 CREATED ON 01/20/83
ENTRIES= 3 LAST ENTRY:11/11/83
LOG BOOK MENU
'?' SHOWS THIS MENU
'QSO' ENTERS A NEW QSO INTO THE LOG
'QRT' EXITS THE LOG PROGRAM
'LIST' RETRIEVES/PRINTS LOG DATA
'FIND' FINDS A CALL IN THE LOG BOOK
'LAST' SHOWS LAST 5 LOG ENTRIES
'EDIT' EDITS A LOG ENTRY
ENTER COMMAND?

Here is the main menu of the LOGBOOK program. This program will store thousands of QSOs on a floppy disk and then retrieve them for printout after they have been sorted into various categories.

designation. For instance, for CP/M you would change 180 as follows: 180 F1$="B:LOGDATA" (assuming you want the data files on drive B:). You would then make a similar change in line 190. However, in the Apple version, the drive designation must be changed in a different way. For Apple, there is a variable in line 110 called DV$. That is now set to ",D1". To designate drive #2 for data files, change it to ",D2".

The program creates two different data files: "LOGINDEX" and "LOGDATA". The index file includes information to facilitate rapid data retrieval. LOGDATA is the full data file, and is apt to become quite large over a period of time. Keep an eye on your disk space to avoid an untimely crash when the disk gets full.

When you first use the LOGBOOK program with a new disk, the data files must be created. The program will detect the lack of data files and will ask you to enter your call, the "volume number" (e.g., volume 1) and the starting date for this particular log. The files will then be set up.

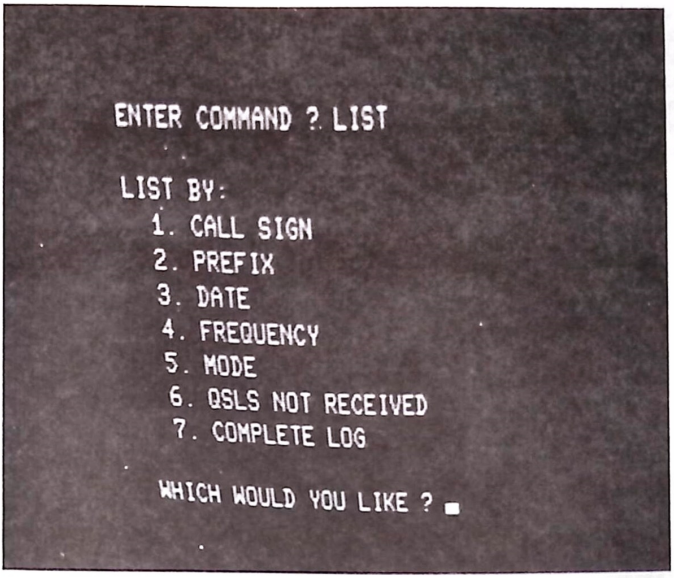
Operating notes

The menu explains how to use the program. The menu, shown in a photograph, offers various options. As with all menu-oriented programs in this book, this one allows you to redisplay the menu by typing a question mark whenever the program says, "ENTER COMMAND."

The two most-used commands in the program are QSO, which you use to enter each contact, and LIST, which allows you to choose various options for data retrieval and printout.

When you enter a QSO, the program prompts you with questions about the standard log entries. If you wish to omit some of the information (such as the ending time of the QSO), simply type a carriage return when asked to enter that item.

In response to the "QSL SENT" query, you have some options. In addition to "Y" or "N" for yes or no, you can send a numeral from 0



The key to the LOGBOOK program's versatility is the LIST function, which allows the operator to print out a complete log, or to select only contacts with a certain station or prefix, or on a certain mode, date, or frequency. In addition, contacts for which QSL cards are needed may be listed separately.

through 9. A "0" means no card sent, but any higher number means that a card was sent and also indicates how many times you have sent a QSL to this station. Or you can hit a carriage return to indicate that no QSL will be sent.

After you have entered the standard information, you can add anything you wish as a "comment," up to the maximum number of characters you have selected for your comment field. As written, the program allows 25 characters, but that can be changed, as we explained earlier.

Probably the most useful command is "LIST," which brings up a secondary menu showing the various ways in which you can retrieve data from the log. When you select the number that matches the option you want, the program searches for the specified data and prints a logsheet including the appropriate log entries.

Sometimes it's possible to get a little confused about when you're supposed to use the main menu commands (like QSO, LIST, or QRT) and when the numbered commands on the secondary LIST menu are appropriate. If the prompt on the screen says, "ENTER COMMAND", you're at the main menu. If you see the secondary menu and the prompt says, "WHICH NUMBER," no command except one of the seven numbers will work.

The handling of data files varies considerably on various brands of computers. On some (for instance, the Apple), it is necessary to open and

close a data file after each read or write operation. Therefore, the time required for the searching and listing functions varies considerably. If it seems that your disk drive is rummaging around for a while, be patient.

When you're finished with the program, be sure to use the "QRT" command rather than just turning the computer off or rebooting. If the program is aborted prematurely, the disk files will not be closed on some computers, and data may be lost.

Modifications for CP/M

To adapt the TRS-80 and CP/M listing for use on CP/M computers running Microsoft BASIC version 5.0 or later (MBASIC.COM), only a few changes are needed:

1. Delete line 25 (the CLEAR statement);
2. Enter your video highlighting codes (if any) at line 60;
3. Delete all TRS-80 clear-screen commands (CLS) and replace them with your system's comparable command. For some terminals, PRINT CHR$(26) will do this. The CLS command is in lines 250, 370 and 1870.
4. If you have a 24 × 80 display, you may wish to center the sign-on message and the menus with additional tabbing (see lines 260–290, 980–1050 and 2480–2570).

LOGBOOK (TRS-80 AND CP/M)

```
10 REM COMPREHENSIVE LOG BOOK PROGRAM (TRS-80)
20 REM BY KC6A AND N6NB;(C) 1983 BY HAYDEN BOOK CO.,INC.
30 CLEAR 1000:PW=70:REM COLUMN WIDTH OF PRINTOUT
40 Ll=66: REM # LINES/PAGE
50 BL$=CHR$ (7):REM NEEDED--DON'T DELETE
60 HI$="":LO$="":REM HIGHLIGHTING CODES
130 E1$="TOO LONG. "
140 E2$=" CHARACTERS MAXIMUM."
150 C1=25: REM LENGTH OF COMMENT FIELD. IF CHANGED, ALSO CHANGE CM$ (LINE 202)
160 C2=49+C1:NW=C1+67
180 F1$="LOGDATA":R1=C2
190 F2$="LOGINDEX":R2=8
201 OPEN "R",1,F1$,R1
202 FIELD 1,8 AS DT$,4 AS UT$,10 AS KL$,3 AS RS$,3 AS RR$,8 AS FQ$,3 AS MD$,4 AS
      PR$,4 AS UE$,1 AS QS$,1 AS QR$,25 AS CM$
204 OPEN "R",2,F2$,R2
205 FIELD 2,6 AS QL$,2 AS P$
206 IF LOF(1)<1 THEN GOTO 3100
210 RN=0: GOSUB 2590
230 G$=KL$: GOSUB 3040:MK$=G$
240 A=20 - LEN (MK$) / 2
250 CLS:PRINT:PRINT HI$
260 PRINT "****************************************"
270 PRINT "                LOG BOOK OF                 "
280 PRINT TAB(A-1)MK$;TAB(39)""
290 PRINT "****************************************"
300 G$=FQ$: GOSUB 3040:CD$=G$
310 G$=CM$: GOSUB 3040:LE$=G$
320 VL=VAL (UT$):LR=VAL (DT$)
330 PRINT LO$:PRINT:PRINT TAB( 6)"VOL ";VL;"  CREATED ON ";FQ$
340 PRINT "# ENTRIES= ";LR;"   LAST ENTRY:";LE$
345 PRINT:CR$="":INPUT "<ENTER> TO CONTINUE";CR$
350 GOSUB 2470: REM MENU
360 PRINT:C$="":INPUT "ENTER COMMAND ";C$
370 IF C$="?" THEN CLS:GOSUB 2470:GOTO 360
380 IF C$="QSO" THEN 450
390 IF C$="QRT" THEN CLOSE:END
400 IF C$="FIND" THEN 740
410 IF C$="LIST" THEN 960
```

```
420 IF C$="EDIT" THEN 1810
430 IF C$="LAST" THEN 890
440 PRINT:PRINT BL$;"INVALID COMMAND. PLEASE TRY AGAIN": GOTO 360
450 REM ADD A NEW QSO
460 GOSUB 2390:LSET KL$=K$:REM GET CALL
470 GOSUB 2220:LSET DT$=DA$:REM GET DATE
480 PRINT "WHAT IS THE STARTING TIME ";: GOSUB 2340:LSET UT$=U$: REM GET TIME
490 A$="":INPUT "WHAT IS THE FREQUENCY ";A$:LSET FQ$=A$
500 IF LEN (A$)>8 THEN PRINT BL$;E1$;8;E2$
510 A$="":INPUT "WHAT IS THE MODE ";A$:LSET MD$=A$
520 IF LEN (A$)>3 THEN PRINT BL$;E1$;3;E2$: GOTO 510
530 A$="":INPUT "WHAT IS THE POWER ";A$:LSET PR$=A$
535 IF LEN(A$)>4 THEN PRINT BL$;E1$;4;E2$:GOTO 530
540 A$="":INPUT "WHAT IS HIS RST ";A$:LSET RS$=A$
550 IF LEN (A$)>3 THEN PRINT BL$;E1$;3;E2$: GOTO 540
560 A$="":INPUT "WHAT RST DID YOU RECEIVE ";A$:LSET RR$=A$
570 IF LEN (A$)>3 THEN PRINT BL$;E1$;3;E2$: GOTO 560
590 PRINT "WHAT IS THE ENDING TIME OF THE QSO ";: GOSUB 2340:LSET UE$=U$
600 A$="":INPUT "IS A QSL BEING SENT (Y/N) ";A$:LSET QS$=A$
610 IF LEN (A$)>1 THEN PRINT BL$;E1$;1;E2$: GOTO 600
620 LSET QR$=""
630 PRINT "ENTER ANY COMMENTS (";C1;" CHARACTERS MAX)"
640 A$="":INPUT A$:LSET CM$=A$
650 IF LEN (A$)>C1 THEN PRINT BL$; LEN (A$) - C1;" CHARACTERS TOO MANY. RETRY.":
    GOTO 640
660 LR=LR + 1:RN=LR: GOSUB 2760: REM WRITE LOG
670 RN=H: GOSUB 2930:IF P=0 THEN 690
680 RN=P: GOSUB 2930:IF P<>0 THEN 680
690 P=LR + 26: GOSUB 2990
700 RN=P:P=0:LSET QL$=KX$: GOSUB 2990
710 RN=0: GOSUB 2590
720 LSET CM$=DA$:LSET DT$= STR$ (LR): GOSUB 2760
730 GOTO 360
740 REM  FIND A CALL & DISPLAY IT
750 GOSUB 2390: GOSUB 800:X=0: REM  GET CALL & DISPLAY HEADER
760 RN=H: GOSUB 2930: IF P=0 THEN 790
770 RN=P: GOSUB 2930: IF KX$=QM$ THEN  GOSUB 840
780 IF P>0 THEN 770
790 IF X=0 THEN PRINT "NOT FOUND"
795 GOTO 360
800 PRINT:PRINT "RCD#   DATE      TIME     MODE RST  QRT "
810 PRINT "        CALL       FREQ     PWR  S/R  QSL"
820 PRINT "---- ---------- -------- ---- ---  ----"
830 RETURN
840 RN=RN - 26: GOSUB 2590: REM  READ LOG
850 PRINT RN; TAB( 6)DT$; TAB( 17)UT$; TAB( 26)MD$; TAB( 31)RS$; TAB( 36)UE$
860 PRINT  TAB( 6)KL$; TAB( 17)FQ$; TAB( 26)PR$; TAB( 31)RR$; TAB( 37)QS$;"/";QR
    $
870 PRINT  TAB( 6)"COMMENTS:";CM$
880 PRINT:X=1:RETURN
890 GOSUB 800: REM  DISPLAY LAST 5 LOG ENTRIES
900 IF LR=0 THEN  GOTO 360
910 N=LR - 4: IF LR<5 THEN N=1
920 FOR RN=N TO LR
930 GOSUB 2590: GOSUB 850
940 NEXT RN
950 GOTO 360
960 REM  LIST LOG ON PRINTER
970 N1=0:L2=L1 + 6
980 PRINT:PRINT "LIST BY:"
990 PRINT "  1. CALL SIGN"
1000 PRINT "  2. PREFIX"
1010 PRINT "  3. DATE"
1020 PRINT "  4. FREQUENCY"
1030 PRINT "  5. MODE"
1035 PRINT "  6. QSLS NOT RECEIVED"
1040 PRINT "  7. COMPLETE LOG"
1050 PRINT:INPUT "WHICH NUMBER";PL
1060 ON PL GOTO 1120,1170,1240,1400,1320,1481,1080
1070 PRINT "INVALID";BL$: GOTO 360
```

```
1080 FOR RN=1 TO LR:REM LIST COMPLETE LOG
1090 GOSUB 2590: GOSUB 1630
1100 NEXT RN
1110 GOTO 790
1120 GOSUB 2390: REM LIST BY CALL
1130 RN=H: GOSUB 2930: IF P=0 THEN 360
1140 RN=P: GOSUB 2930: IF KX$=QM$ THEN RN=RN - 26: GOSUB 2590: GOSUB 1630
1150 IF P>0 THEN 1140
1160 GOTO 790
1170 PX$="":INPUT "WHAT IS THE PREFIX ";PX$:LX= LEN (PX$):REM LIST BY PREFIX
1180 IF PX$="" THEN  PRINT BL$;"ERROR": GOTO 360
1190 FOR RN=1 TO LR
1200 GOSUB 2590:G$=KL$:GOSUB 3040
1210 IF LEN(G$)>=LX THEN  IF LEFT$ (G$,LX)=PX$ THEN  GOSUB 1630
1220 NEXT RN
1230 GOTO 790
1240 DA$="":INPUT "WHAT IS THE STARTING DATE (MM/DD/YY) ";DA$: GOSUB 2240
1242 SD$=RIGHT$(DA$,2)+"/"+LEFT$(DA$,5)
1250 DA$="":INPUT "WHAT IS THE ENDING DATE (MM/DD/YY) ";DA$: GOSUB 2240
1252 ED$=RIGHT$(DA$,2)+"/"+LEFT$(DA$,5)
1260 IF SD$>ED$ THEN  PRINT BL$"ERROR": GOTO 360
1270 FOR RN=1 TO LR
1280 GOSUB 2590:X$=RIGHT$(DT$,2)+"/"+LEFT$(DT$,5)
1290 IF X$>=SD$ THEN IF X$<=ED$ THEN GOSUB 1630
1300 NEXT RN
1310 GOTO 790
1320 SM$="":INPUT "WHAT IS THE MODE ";SM$:REM LIST BY MODE
1330 IF LEN (SM$)>3 THEN  PRINT BL$;E1$;3;E2$: GOTO 1320
1340 IF LEN (SM$)<3 THEN  FOR J= LEN (SM$) + 1 TO 3:SM$=SM$ + " ": NEXT J
1350 FOR RN=1 TO LR
1360 GOSUB 2590
1370 IF SM$=MD$ THEN  GOSUB 1630
1380 NEXT RN
1390 GOTO 790
1400 INPUT "WHAT IS THE LOWER FREQUENCY LIMIT ";LF:REM LIST BY FREQ
1410 INPUT "WHAT IS THE UPPER FREQUENCY LIMIT ";UF
1420 IF UF<LF THEN  PRINT BL$;"ERROR": GOTO 360
1430 FOR RN=1 TO LR
1440 GOSUB 2590
1450 F= VAL (FQ$)
1460 IF F>=LF THEN  IF F<=UF THEN  GOSUB 1630
1470 NEXT RN
1480 GOTO 790
1481 FOR RN=1 TO LR:REM LIST BY QSL'D
1482 GOSUB 2590
1483 IF QR$="Y" THEN 1485
1484 IF QS$<>" " THEN IF QS$<>"N" THEN IF QS$<>"0" THEN GOSUB 1630
1485 NEXT RN:GOTO 790
1490 N1=N1 + 1: REM  NEW PAGE
1500 IF L2>=L1 + 6 THEN 1520
1510 FOR J=L2 + 1 TO L1 + 6: LPRINT:NEXT J
1520 LPRINT  TAB(29)"LOG OF ";MK$;TAB(55)"PAGE ";N1
1530 LPRINT TAB(33)"VOL ";VL
1540 LPRINT
1550 LPRINT TAB(50)"HIS MY       QSL"
1560 LPRINT "RCD#   DATE    TIME     CALL      FREQ   MODE PWR  RST RST QRT  S/R";
1570 IF NW<=PW THEN LPRINT "     COMMENTS";
1580 LPRINT
1590 LPRINT "---- -------- ---- ---------- -------- ---  ---- --- --- ---- ---";
1600 IF NW<=PW THEN LPRINT "  ---------------";
1610 LPRINT
1620 L2=12: RETURN
1630 IF L2 + 3>L1 - 6 THEN  GOSUB 1490: REM  NEW PAGE
1640 LPRINT RN;TAB(6)DT$;TAB(15)UT$;TAB(20)KL$;TAB(31)FQ$;TAB(40)MD$;TAB(45)PR$;
1650 LPRINT TAB(50)RS$;TAB(54)RR$;TAB(58)UE$;TAB(63)QS$;"/";QR$;
1760 IF NW<=PW THEN  LPRINT "  ";CM$:L2=L2 + 1: GOTO 1800
1770 LPRINT
1780 LPRINT TAB(6)"COMMENTS: ";CM$
1790 LPRINT:L2=L2+3:X=1
1800 RETURN
```

```
1810 PRINT:PRINT "ENTER THE RECORD # TO BE EDITED OR"
1820 A$="": INPUT "<CR> TO EDIT THE LAST LOG ENTRY ";A$
1830 IF A$="" THEN RN=LR: GOTO 1850
1840 RN= VAL (A$)
1850 IF RN>LR THEN 2210
1860 IF RN<1 THEN 2210
1870 CLS:GOSUB 2590: GOSUB 800: GOSUB 850
1880 A$="":INPUT "EDIT THIS (Y/N) ";A$
1890 IF A$<>"Y" THEN 360
1900 PRINT "ENTER <CR> TO LEAVE AN ITEM UNCHANGED."
1910 PRINT "ENTER A NEW VALUE TO CHANGE IT."
1920 PRINT:PRINT "DATE: ";DT$;:DA$="":INPUT DA$: IF DA$<>"" THEN  GOSUB 2240:LSE
     T DT$=DA$
1930 PRINT "STARTING TIME: ";UT$;:U$="":INPUT U$: IF U$<>"" THEN  GOSUB 2350:LSE
     T UT$=U$
1940 PRINT "FREQUENCY: ";FQ$;:A$="":INPUT A$: IF A$="" THEN 1970
1950 IF LEN (A$)>8 THEN  PRINT BL$;E1$;8;E2$: GOTO 1940
1960 LSET FQ$=A$
1970 PRINT "MODE: ";MD$;:A$="":INPUT A$: IF A$="" THEN 2000
1980 IF LEN (A$)>3 THEN  PRINT BL$;E1$;3;E2$: GOTO 1970
1990 LSET MD$=A$
2000 PRINT "POWER: ";PR$;:A$="":INPUT A$: IF A$="" THEN 2030
2010 IF LEN (A$)>4 THEN  PRINT BL$;E1$;4;E2$: GOTO 2000
2020 LSET PR$=A$
2030 PRINT "RST SENT: ";RS$;:A$="":INPUT A$: IF A$="" THEN 2060
2040 IF LEN (A$)>3 THEN  PRINT E1$;3;E2$: GOTO 2030
2050 LSET RS$=A$
2060 PRINT "MY RST: ";RR$;:A$="":INPUT A$: IF A$="" THEN 2090
2070 IF LEN (A$)>3 THEN  PRINT E1$;3;E2$: GOTO 2060
2080 LSET RR$=A$
2090 PRINT "ENDING TIME: ";UE$;:U$="":INPUT U$: IF U$<>"" THEN  GOSUB 2350:LSET
     UE$=U$
2100 PRINT "QSL SENT: ";QS$;:A$="":INPUT A$: IF A$="" THEN 2130
2110 IF LEN (A$)<>1 THEN  PRINT BL$;E1$;1;E2$: GOTO 2100
2120 LSET QS$=A$
2130 PRINT "QSL RCVD: ";QR$;:A$="":INPUT A$: IF A$="" THEN 2160
2140 IF LEN (A$)<>1 THEN  PRINT BL$;E1$;1;E2$: GOTO 2130
2150 LSET QR$=A$
2160 PRINT "COMMENTS: ";CM$:A$="":INPUT "          ";A$: IF A$="" THEN 2190
2170 IF LEN (A$)>C1 THEN  PRINT BL$;E1$;C1;E2$: GOTO 2160
2180 LSET CM$=A$
2190 A$="":INPUT "OK (Y/N) ";A$: IF A$<>"Y" THEN 360
2200 GOSUB 2760: GOTO 360
2210 PRINT BL$;"INVALID RECORD #": GOTO 360
2220 REM  GET DATE
2230 DA$="":INPUT "WHAT IS THE DATE (MM/DD/YY) ";DA$
2240 IF LEN (DA$)<6 THEN 2330
2250 IF MID$ (DA$,3,1)="/" THEN 2280
2260 IF MID$ (DA$,2,1)<>"/" THEN 2330
2270 DA$="0" + DA$
2280 IF MID$ (DA$,6,1)="/" THEN 2310
2290 IF MID$ (DA$,5,1)<>"/" THEN 2330
2300 DA$= LEFT$ (DA$,3) + "0" +  RIGHT$ (DA$,4)
2310 IF LEN (DA$)<>8 THEN 2330
2320 RETURN
2330 PRINT BL$;"INVALID. RE-ENTER.": GOTO 2220
2340 U$="":INPUT "(HHMM) ";U$: REM  GET TIME
2350  IF LEN (U$)>4 THEN  PRINT BL$;E1$;4;E2$;" RE-ENTER": GOTO 2340
2360 IF U$="" THEN  RETURN
2370 IF LEN (U$)<4 THEN  FOR J= LEN (U$) + 1 TO 4:U$="0" + U$: NEXT J
2380 RETURN
2390 REM  GET CALL SIGN
2400 K$="":INPUT "WHAT IS THE CALL SIGN ";K$
2410 IF LEN (K$)>10 THEN  PRINT BL$;E1$;10;E2$: GOTO 2400
2420 IF LEN (K$)<4 THEN 2460
2422 FOR J=1 TO LEN(K$):IF MID$(K$,J,1)="/" THEN J=J-1:GOTO 2425
2423 NEXT J
2425 IF J>6 THEN J=6
2426 KX$=MID$(K$,1,J)
2430 H= ASC ( MID$ (K$,4,1)) -  ASC ("A") + 1
```

```
2440 IF H<1 THEN 2460
2450 IF H<=26 THEN  RETURN
2460 PRINT BL$;"INVALID CALL": GOTO 2400
2470 REM  MENU
2480 PRINT:PRINT HI$:PRINT "****************************************"
2485 PRINT "              LOG BOOK MENU                 "
2490 PRINT "                                            "
2500 PRINT " '?' SHOWS THIS MENU                       "
2510 PRINT " 'QSO' ENTERS A NEW QSO INTO THE LOG       "
2520 PRINT " 'QRT' EXITS THE LOG PROGRAM               "
2530 PRINT " 'LIST' RETRIEVES/PRINTS LOG DATA          "
2540 PRINT " 'FIND' FINDS A CALL IN THE LOG BOOK       "
2550 PRINT " 'LAST' SHOWS LAST 5 LOG ENTRIES           "
2560 PRINT " 'EDIT' EDITS A LOG ENTRY                  "
2570 PRINT "****************************************"
2580 PRINT LO$:PRINT:RETURN
2590 REM  READ LOG
2600 GET 1,RN+1
2750 RETURN
2760 REM  WRITE LOG
2780 PUT 1,RN+1
2920 RETURN
2930 REM  READ F2
2940 GET 2,RN
2950 P=CVI(P$)
2960 G$=QL$:GOSUB 3040:QM$=G$
2980 RETURN
2990 REM  WRITE F2
3000 LSET P$=MKI$(P)
3010 PUT 2,RN:RETURN
3040 REM  REMOVE TRAILING BLANKS FROM G$
3050 FOR J=1 TO  LEN (G$): IF MID$ (G$,J,1)=" " THEN 3070
3060 NEXT J
3070 IF J=1 THEN G$="": RETURN
3080 G$= MID$ (G$,1,J - 1)
3090 RETURN
3100 PRINT:PRINT  TAB( 9)"INITIALIZING LOG FILES"
3120 PRINT :MK$="":INPUT "WHAT IS YOUR CALL ";MK$
3130 PRINT:INPUT "WHAT VOLUME NUMBER IS THIS LOG ";VL
3140 PRINT:GOSUB 2220:CD$=DA$
3150 LSET DT$="0":LR=0
3160 LSET KL$=MK$:LSET FQ$=CD$
3170 LSET UT$=STR$(VL):LSET CM$=CD$
3180 RN=0: GOSUB 2760
3190 P=0:LSET QL$=""
3200 FOR RN=1 TO 26
3210 GOSUB 2990
3220 NEXT RN
3230 GOTO 230
```

LOGBOOK (Apple)

```
10  REM  COMPREHENSIVE LOG BOOK PROGRAM (APPLE)
20  REM  BY KC6A AND N6NB;(C) 1983 BY HAYDEN BOOK CO.,INC.
30 PW = 70: REM  # COLS FOR HARD COPY
40 L1 = 66: REM  # LINES/PAGE
50 D$ =  CHR$ (4):BL$ =  CHR$ (7)
60 LP$ = D$ + "PR#1":SCN$ = D$ + "PR#0"
70 OP$ = D$ + "OPEN ":RD$ = D$ + "READ ":WR$ = D$ + "WRITE ":CL$ = D$ + "CLOSE "
110 DL$ = D$ + "DELETE ":DV$ = ",D1":E1$ = "TOO LONG. ":E2$ = " CHARACTERS MAX"
150 C1 = 25: REM  LENGTH OF COMMENT FIELD
160 C2$ =  STR$ (50 + C1):NW = C1 + 67
180 F1$ = "LOGDATA":R1$ = ",L" + C2$:F2$ = "LOGINDEX":R2$ = ",L12"
200  ONERR  GOTO 3100
210 RN = 0: GOSUB 2590
220  POKE 216,0
230 G$ = KL$: GOSUB 3040:MK$ = G$
240 A = 20 -  LEN (MK$) / 2
250  HOME : PRINT : INVERSE
260  PRINT  SPC( 39)""
270  PRINT "                LOG BOOK OF                "
280  PRINT  SPC( A)MK$; SPC( 39 - A -  LEN (MK$))""
290  PRINT  SPC( 39)""
300 G$ = FQ$: GOSUB 3040:CD$ = G$
310 G$ = CM$: GOSUB 3040:LE$ = G$
320 VL =  VAL (UT$):LR =  VAL (DT$)
330  NORMAL : PRINT : PRINT  TAB( 6)"VOL ";VL;"  CREATED ON ";FQ$
340  PRINT "# ENTRIES= ";LR;"   LAST ENTRY:";LE$
350  GOSUB 2470: REM  MENU
360  PRINT : INPUT "ENTER COMMAND? ";C$
370  IF C$ = "?" THEN  HOME : GOSUB 2470: GOTO 360
380  IF C$ = "QSO" THEN 450
390  IF C$ = "QRT" THEN  END
400  IF C$ = "FIND" THEN 740
410  IF C$ = "LIST" THEN 960
420  IF C$ = "EDIT" THEN 1810
430  IF C$ = "LAST" THEN 890
440  PRINT : PRINT BL$;"INVALID COMMAND. PLEASE TRY AGAIN": GOTO 360
450  REM  ADD A NEW QSO
460  GOSUB 2390:KL$ = K$: REM  GET CALL
470  GOSUB 2220:DT$ = DA$: REM  GET DATE
480  PRINT "WHAT IS THE STARTING TIME ";: GOSUB 2340:UT$ = U$: REM  GET TIME
490  INPUT "WHAT IS THE FREQUENCY? ";FQ$
500  IF  LEN (FQ$) > 8 THEN  PRINT BL$;E1$;8;E2$
510  INPUT "WHAT IS THE MODE? ";MD$
520  IF  LEN (MD$) > 3 THEN  PRINT BL$;E1$;3;E2$: GOTO 510
530  INPUT "WHAT IS THE POWER? ";PR$
535  IF  LEN (PR$) > 4 THEN  PRINT BL$;E1$;4;E2$: GOTO 530
540  INPUT "WHAT IS HIS RST? ";RS$
550  IF  LEN (RS$) > 3 THEN  PRINT BL$;E1$;3;E2$: GOTO 540
560  INPUT "WHAT RST DID YOU RECEIVE? ";RR$
570  IF  LEN (RR$) > 3 THEN  PRINT BL$;E1$;3;E2$: GOTO 560
590  PRINT "WHAT IS THE ENDING TIME OF THE QSO ";: GOSUB 2340:UE$ = U$
600  INPUT "IS A QSL BEING SENT (Y/N)? ";QS$
610  IF  LEN (QS$) > 1 THEN  PRINT BL$;E1$;1;E2$: GOTO 600
620 QR$ = ""
630  PRINT "ENTER ANY COMMENTS (";C1;" CHARACTERS MAX)"
640  INPUT CM$
650  IF  LEN (CM$) > C1 THEN  PRINT BL$; LEN (CM$) - C1;" CHARACTERS TOO MANY. RET
RY.": GOTO 640
660 LR = LR + 1:RN = LR: GOSUB 2760: REM  WRITE LOG
670 RN = H: GOSUB 2930: IF P = 0 THEN 690
680 RN = P: GOSUB 2930: IF P <  > 0 THEN 680
690 P = LR + 26: GOSUB 2990
700 RN = P:P = 0:QL$ = KX$: GOSUB 2990
710 RN = 0: GOSUB 2590
720 CM$ = DA$:DT$ =  STR$ (LR): GOSUB 2760
730  GOTO 360
740  REM  FIND A CALL & DISPLAY IT
```

```
750  GOSUB 2390: GOSUB 800:X = 0: REM  CALL/DISPLAY HEADER
760 RN = H: GOSUB 2930: IF P = 0 THEN 790
770 RN = P: GOSUB 2930: IF KX$ = QL$ THEN  GOSUB 840
780  IF P > 0 THEN 770
790  IF X = 0 THEN  PRINT "NOT FOUND"
795  GOTO 360
800  PRINT : PRINT "RCD#   DATE      TIME    MODE RST  QRT "
810  PRINT "       CALL      FREQ    PWR  S/R  QSL"
820  PRINT "---- ---------- -------- ---- ---  ----"
830  RETURN
840 RN = RN - 26: GOSUB 2590: REM  READ LOG
850  PRINT RN; TAB( 6)DT$; TAB( 17)UT$; TAB( 26)MD$; TAB( 31)RS$; TAB( 36)UE$
860  PRINT  TAB( 6)KL$; TAB( 17)FQ$; TAB( 26)PR$; TAB( 31)RR$; TAB( 37)QS$;"/";QR$
870  PRINT  TAB( 6)"COMMENTS:";CM$
880  PRINT :X = 1: RETURN
890  GOSUB 800: REM  DISPLAY LAST 5 LOG ENTRIES
900  IF LR = 0 THEN  GOTO 360
910 N = LR - 4: IF LR < 5 THEN N = 1
920  FOR RN = N TO LR
930  GOSUB 2590: GOSUB 850
940  NEXT RN
950  GOTO 360
960  REM  LIST LOG ON PRINTER
970 N1 = 0:L2 = L1 + 6
980  PRINT : PRINT "LIST BY:"
990  PRINT "  1. CALL SIGN"
1000  PRINT "  2. PREFIX"
1010  PRINT "  3. DATE"
1020  PRINT "  4. FREQUENCY"
1030  PRINT "  5. MODE"
1035  PRINT "  6. QSL'S NOT RECEIVED"
1040  PRINT "  7. COMPLETE LOG"
1050  PRINT : INPUT "WHICH NUMBER?";PL
1060  ON PL GOTO 1120,1170,1240,1400,1320,1481,1080
1070  PRINT "INVALID";BL$: GOTO 360
1080  FOR RN = 1 TO LR
1090  GOSUB 2590: GOSUB 1630
1100  NEXT RN
1110  GOTO 790
1120  GOSUB 2390
1130 RN = H: GOSUB 2930: IF P = 0 THEN 360
1140 RN = P: GOSUB 2930: IF KX$ = QL$ THEN RN = RN - 26: GOSUB 2590: GOSUB 1630
1150  IF P > 0 THEN 1140
1160  GOTO 790
1170  INPUT "WHAT IS THE PREFIX? ";PX$:LX =  LEN (PX$)
1180  IF PX$ = "" THEN  PRINT BL$;"ERROR": GOTO 360
1190  FOR RN = 1 TO LR
1200  GOSUB 2590:G$ = KL$: GOSUB 3040
1210  IF  LEN (G$) >  = LX THEN  IF  LEFT$ (G$,LX) = PX$ THEN  GOSUB 1630
1220  NEXT RN
1230  GOTO 790
1240  INPUT "WHAT IS THE STARTING DATE (MM/DD/YY)? ";DA$: GOSUB 2240
1242 SD$ =  RIGHT$ (DA$,2) + "/" +  LEFT$ (DA$,5)
1250  INPUT "WHAT IS THE ENDING DATE (MM/DD/YY)? ";DA$: GOSUB 2240
1252 ED$ =  RIGHT$ (DA$,2) + "/" +  LEFT$ (DA$,5)
1260  IF SD$ > ED$ THEN  PRINT BL$"ERROR": GOTO 360
1270  FOR RN = 1 TO LR
1280  GOSUB 2590:X$ =  RIGHT$ (DT$,2) + "/" +  LEFT$ (DT$,5)
1290  IF X$ >  = SD$ THEN  IF X$ <  = ED$ THEN  GOSUB 1630
1300  NEXT RN
1310  GOTO 790
1320  INPUT "WHAT IS THE MODE? ";SM$
1323  IF QR$ = "Y" THEN 1425
1330  IF  LEN (SM$) > 3 THEN  PRINT BL$;E1$;3;E2$: GOTO 1320
1340  IF  LEN (SM$) < 3 THEN  FOR J =  LEN (SM$) + 1 TO 3:SM$ = SM$ + " ": NEXT J
1350  FOR RN = 1 TO LR
1360  GOSUB 2590
1370  IF SM$ = MD$ THEN  GOSUB 1630
1380  NEXT RN
1390  GOTO 790
```

```
1400  INPUT "WHAT IS THE LOWER FREQUENCY LIMIT? ";LF
1410  INPUT "WHAT IS THE UPPER FREQUENCY LIMIT? ";UF
1420  IF UF < LF THEN  PRINT BL$;"ERROR": GOTO 360
1430  FOR RN = 1 TO LR
1440  GOSUB 2590
1450 F =  VAL (FQ$)
1460  IF F >  = LF THEN  IF F <  = UF THEN  GOSUB 1630
1470  NEXT RN
1480  GOTO 790
1481  FOR RN = 1 TO LR: GOSUB 2590: IF QR$ = "Y" THEN 1485
1484  IF QS$ <  > " " THEN  IF QS$ <  > "N" THEN  IF QS$ <  > "0" THEN  GOSUB 1630
1485  NEXT RN: GOTO 790
1490 N1 = N1 + 1: REM  NEW PAGE
1500  IF L2 >  = L1 + 6 THEN 1520
1510  FOR J = L2 + 1 TO L1 + 6: PRINT : NEXT J
1520  PRINT  SPC( 28)"LOG OF ";MK$; SPC( 25 -  LEN (MK$))"PAGE ";N1
1530  PRINT  SPC( 32)"VOL ";VL
1540  PRINT
1550  PRINT  SPC( 49)"HIS MY       QSL"
1560  PRINT "RCD#   DATE   TIME    CALL      FREQ   MODE PWR  RST RST QRT  S/R";
1570  IF NW <  = PW THEN  PRINT "     COMMENTS";
1580  PRINT
1590  PRINT "---- -------- ---- ---------- -------- ---  ---- --- --- ---- ---";
1600  IF NW <  = PW THEN  PRINT "  ---------------";
1610  PRINT
1620 L2 = 12: RETURN
1630  PRINT LP$: IF L2 + 3 > L1 - 6 THEN  GOSUB 1490: REM  NEW PAGE
1640  PRINT RN;
1650  PRINT  SPC( 5 -  LEN ( STR$ (RN)))DT$;
1660  PRINT  SPC( 9 -  LEN (DT$))UT$;
1670  PRINT  SPC( 5 -  LEN (UT$))KL$;
1680  PRINT  SPC( 11 -  LEN (KL$))FQ$;
1690  PRINT  SPC( 9 -  LEN (FQ$))MD$;
1700  PRINT  SPC( 5 -  LEN (MD$))PR$;
1710  PRINT  SPC( 5 -  LEN (PR$))RS$;
1720  PRINT  SPC( 4 -  LEN (RS$))RR$;
1730  PRINT  SPC( 4 -  LEN (RR$))UE$;
1740  PRINT  SPC( 5 -  LEN (UE$))QS$;
1750  PRINT  SPC( 1 -  LEN (QS$))"/";QR$;
1760  IF NW <  = PW THEN  PRINT "  ";CM$:L2 = L2 + 1: GOTO 1800
1770  PRINT
1780  PRINT  SPC( 5)"COMMENTS: ";CM$
1790  PRINT :L2 = L2 + 3:X = 1
1800  PRINT SCN$: RETURN
1810  PRINT : PRINT "ENTER THE RECORD # TO BE EDITED OR"
1820  INPUT "<CR> TO EDIT THE LAST LOG ENTRY? ";A$
1830  IF A$ = "" THEN RN = LR: GOTO 1850
1840 RN =  VAL (A$)
1850  IF RN > LR THEN 2210
1860  IF RN < 1 THEN 2210
1870  HOME : GOSUB 2590: GOSUB 800: GOSUB 850
1880  INPUT "EDIT THIS (Y/N)? ";A$
1890  IF A$ <  > "Y" THEN 360
1900  PRINT "ENTER <CR> TO LEAVE AN ITEM UNCHANGED."
1910  PRINT "ENTER A NEW VALUE TO CHANGE IT."
1920  PRINT : PRINT "DATE: ";DT$;: INPUT DA$: IF DA$ <  > "" THEN  GOSUB 2240:DT$
= DA$
1930  PRINT "STARTING TIME: ";UT$;: INPUT U$: IF U$ <  > "" THEN  GOSUB 2350:UT$ =
 U$
1940  PRINT "FREQUENCY: ";FQ$;: INPUT A$: IF A$ = "" THEN 1970
1950  IF  LEN (A$) > 8 THEN  PRINT BL$;E1$;8;E2$: GOTO 1940
1960 FQ$ = A$
1970  PRINT "MODE: ";MD$;: INPUT A$: IF A$ = "" THEN 2000
1980  IF  LEN (A$) > 3 THEN  PRINT BL$;E1$;3;E2$: GOTO 1970
1990 MD$ = A$
2000  PRINT "POWER: ";PR$;: INPUT A$: IF A$ = "" THEN 2030
2010  IF  LEN (A$) > 4 THEN  PRINT BL$;E1$;4;E2$: GOTO 2000
2020 PR$ = A$
2030  PRINT "RST SENT: ";RS$;: INPUT A$: IF A$ = "" THEN 2060
2040  IF  LEN (A$) > 3 THEN  PRINT E1$;3;E2$: GOTO 2030
```

```
2050 RS$ = A$
2060  PRINT "MY RST: ";RR$;: INPUT A$: IF A$ = "" THEN 2090
2070  IF  LEN (A$) > 3 THEN  PRINT E1$;3;E2$: GOTO 2060
2080 RR$ = A$
2090  PRINT "ENDING TIME: ";UE$;: INPUT U$: IF U$ <  > "" THEN  GOSUB 2350:UE$ = U
$
2100  PRINT "QSL SENT: ";QS$;: INPUT A$: IF A$ = "" THEN 2130
2110  IF  LEN (A$) <  > 1 THEN  PRINT BL$;E1$;1;E2$: GOTO 2100
2120 QS$ = A$
2130  PRINT "QSL RCVD: ";QR$;: INPUT A$: IF A$ = "" THEN 2160
2140  IF  LEN (A$) <  > 1 THEN  PRINT BL$;E1$;1;E2$: GOTO 2130
2150 QR$ = A$
2160  PRINT "COMMENTS: ";CM$: INPUT "        ? ";A$: IF A$ = "" THEN 2190
2170  IF  LEN (A$) > C1 THEN  PRINT BL$;E1$;C1;E2$: GOTO 2160
2180 CM$ = A$
2190  INPUT "OK (Y/N)? ";A$: IF A$ <  > "Y" THEN 360
2200  GOSUB 2760: GOTO 360
2210  PRINT BL$;"INVALID RECORD #": GOTO 360
2220  REM  GET DATE
2230  INPUT "WHAT IS THE DATE (MM/DD/YY)? ";DA$
2240  IF  LEN (DA$) < 6 THEN 2330
2250  IF  MID$ (DA$,3,1) = "/" THEN 2280
2260  IF  MID$ (DA$,2,1) <  > "/" THEN 2330
2270 DA$ = "0" + DA$
2280  IF  MID$ (DA$,6,1) = "/" THEN 2310
2290  IF  MID$ (DA$,5,1) <  > "/" THEN 2330
2300 DA$ =  LEFT$ (DA$,3) + "0" +  RIGHT$ (DA$,4)
2310  IF  LEN (DA$) <  > 8 THEN 2330
2320  RETURN
2330  PRINT BL$;"INVALID. RE-ENTER.": GOTO 2220
2340  INPUT "(HHMM)? ";U$: REM  GET TIME
2350  IF  LEN (U$) > 4 THEN  PRINT BL$;E1$;4;E2$;" RE-ENTER": GOTO 2340
2360  IF U$ = "" THEN  RETURN
2370  IF  LEN (U$) < 4 THEN  FOR J =  LEN (U$) + 1 TO 4:U$ = "0" + U$: NEXT J
2380  RETURN
2390  REM  GET CALL SIGN
2400  INPUT "WHAT IS THE CALL SIGN? ";K$
2410  IF  LEN (K$) > 10 THEN  PRINT BL$;E1$;10;E2$: GOTO 2400
2420  IF  LEN (K$) < 4 THEN 2460
2422  FOR J = 1 TO  LEN (K$): IF  MID$ (K$,J,1) = "/" THEN J = J - 1: GOTO 2425
2423  NEXT J
2425  IF J > 6 THEN J = 6
2426 KX$ =  MID$ (K$,1,J)
2430 H =  ASC ( MID$ (K$,4,1)) -  ASC ("A") + 1
2440  IF H < 1 THEN 2460
2450  IF H <  = 26 THEN  RETURN
2460  PRINT BL$;"INVALID CALL": GOTO 2400
2470  REM  MENU
2480  PRINT : INVERSE : PRINT "             LOG BOOK MENU              "
2490  PRINT  SPC( 39)""
2500  PRINT " '?' SHOWS THIS MENU                    "
2510  PRINT " 'QSO' ENTERS A NEW QSO INTO THE LOG    "
2520  PRINT " 'QRT' EXITS THE LOG PROGRAM            "
2530  PRINT " 'LIST' RETRIEVES/PRINTS LOG DATA       "
2540  PRINT " 'FIND' FINDS A CALL IN THE LOG BOOK    "
2550  PRINT " 'LAST' SHOWS LAST 5 LOG ENTRIES        "
2560  PRINT " 'EDIT' EDITS A LOG ENTRY               "
2570  PRINT  SPC( 39)""
2580  NORMAL : PRINT : RETURN
2590  REM  READ LOG
2600  PRINT OP$;F1$;R1$;DV$
2610  PRINT RD$;F1$;",R";RN
2620  INPUT A$: PRINT CL$
2630 DT$ =  LEFT$ (A$,8)
2640 UT$ =  MID$ (A$,9,4)
2650 KL$ =  MID$ (A$,13,10)
2660 RS$ =  MID$ (A$,23,3)
2670 RR$ =  MID$ (A$,26,3)
2680 FQ$ =  MID$ (A$,29,8)
2690 MD$ =  MID$ (A$,37,3)
```

```
2700 PR$ =  MID$ (A$,40,4)
2710 UE$ =  MID$ (A$,44,4)
2720 QS$ =  MID$ (A$,48,1)
2730 QR$ =  MID$ (A$,49,1)
2740 CM$ =  RIGHT$ (A$,25)
2750  RETURN
2760  REM  WRITE LOG
2770  PRINT OP$;F1$;R1$;DV$
2780  PRINT WR$;F1$;",R";RN
2790  PRINT DT$;
2800  PRINT  SPC( 8 -  LEN (DT$))UT$;
2810  PRINT  SPC( 4 -  LEN (UT$))KL$;
2820  PRINT  SPC( 10 -  LEN (KL$))RS$;
2830  PRINT  SPC( 3 -  LEN (RS$))RR$;
2840  PRINT  SPC( 3 -  LEN (RR$))FQ$;
2850  PRINT  SPC( 8 -  LEN (FQ$))MD$;
2860  PRINT  SPC( 3 -  LEN (MD$))PR$;
2870  PRINT  SPC( 4 -  LEN (PR$))UE$;
2880  PRINT  SPC( 4 -  LEN (UE$))QS$;
2890  PRINT  SPC( 1 -  LEN (QS$))QR$;
2900  PRINT  SPC( 1 -  LEN (QR$))CM$; SPC( 25 -  LEN (CM$))""
2910  PRINT CL$
2920  RETURN
2930  REM  READ F2
2940  PRINT OP$;F2$;R2$;DV$
2950  PRINT RD$;F2$;",R";RN
2960  INPUT QL$: INPUT P
2970  PRINT CL$
2980  RETURN
2990  REM  WRITE F2
3000  PRINT OP$;F2$;R2$;DV$
3010  PRINT WR$;F2$;",R";RN
3020  PRINT QL$: PRINT P
3030  PRINT CL$: RETURN
3040  REM  REMOVE TRAILING BLANKS FROM G$
3050  FOR J = 1 TO  LEN (G$): IF  MID$ (G$,J,1) = " " THEN 3070
3060  NEXT J
3070  IF J = 1 THEN G$ = "": RETURN
3080 G$ =  MID$ (G$,1,J - 1)
3090  RETURN
3100  PRINT : PRINT  TAB( 9)"INITIALIZING LOG FILES"
3110  POKE 216,0
3120  PRINT : INPUT "WHAT IS YOUR CALL? ";MK$
3130  PRINT : INPUT "WHAT VOLUME NUMBER IS THIS LOG? ";VL
3140  PRINT : GOSUB 2220:CD$ = DA$
3150 DT$ = "0":LR = 0
3160 KL$ = MK$:FQ$ = CD$
3170 UT$ =  STR$ (VL):CM$ = CD$
3180 RN = 0: GOSUB 2760
3190 P = 0:QL$ = ""
3200  FOR RN = 1 TO 26
3210  GOSUB 2990
3220  NEXT RN
3230  GOTO 220
```

LOGBOOK (Commodore 64)

```
10 REM COMPREHENSIVE LOG BOOK PROGRAM (C64)
20 REM BY KC6A AND N6NB;(C) 1983 BY HAYDEN BOOK CO.,INC.
25 OPEN130,2,2,CHR$(136):REM RS232 PRINTER CHANNEL
26 OPEN15,8,15:C7$=CHR$(147):PRINT C7$:REM CLEAR SCREEN
28 DIM QT$(1500),PT(1500):REM # QSOS PERMITTED ON ONE DATA DISK
30 PW=70:REM COLUMN WIDTH OF PRINTOUT
```

```
40 L1=66:REM # LINES/PAGE
60 BL$=CHR$(7):HI$=CHR$(18):LO$=CHR$(146):REM BELL AND HIGHLIGHTING CODES
130 E1$="TOO LONG. ":E2$=" CHARACTERS MAXIMUM."
150 C1=25:REM LENGTH OF COMMENT FIELD.
160 C2=61+C1:NW=C1+67
180 F1$="0:LOGDATA":F2$="0:LOGINDEX"
195 PRINT:PRINT:PRINT "  ";HI$;"                                              "
196 PRINT "  ";HI$;"   COMPREHENSIVE LOG BOOK PROGRAM   "
197 PRINT "  ";HI$;"                                                ":PRINT:PRINT
201 A$="":PRINT "DO YOU WANT TO CREATE A NEW LOG FILE";:INPUT A$
202 IF A$="Y" THEN 3100
205 OPEN 3,8,3,F1$:GOSUB 6000
206 PRINT:PRINT TAB(9)"READING DATA FILES..."
210 RN=0:GOSUB 2590
230 G$=KL$:GOSUB 3040:MK$=G$
240 A=INT(20-LEN(MK$)/2)
250 PRINT C7$:PRINT:PRINT:PRINT:PRINT
270 PRINT TAB(14)"LOG BOOK OF":PRINT TAB(A)MK$:PRINT
300 G$=FQ$:GOSUB 3040:CD$=G$
310 G$=CM$:GOSUB 3040:LE$=G$
320 VL=VAL(UT$):LR=VAL(DT$)
330 PRINT LO$:PRINT:PRINT TAB(6)"VOL ";VL;"  CREATED ON ";FQ$
335 PRINT:PRINT " # ENTRIES=";LR;"   LAST ENTRY=";LE$
340 OPEN4,8,4,F2$+",S,R":GOSUB 6000
341 FOR J=1 TO LR+26:INPUT#4,QT$(J),PT(J):NEXT J:CLOSE4
350 GOSUB 2470:REM MENU
360 PRINT:INPUT "ENTER COMMAND ";C$
370 IF C$="?" THEN PRINT C7$:GOSUB 2470:GOTO 360
380 IF C$="QSO" THEN 450
390 IF C$<>"QRT" THEN 400
392 CLOSE3:GOSUB 3500:CLOSE15:FOR J=1TO 1000:PRINT;:NEXT J:CLOSE130:END
400 IF C$="FIND" THEN 740
410 IF C$="LIST" THEN 960
420 IF C$="EDIT" THEN 1810
430 IF C$="LAST" THEN 890
440 PRINT:PRINT BL$;"INVALID COMMAND. PLEASE TRY AGAIN":GOTO 360
450 REM ADD A NEW QSO
460 GOSUB 2390:KL$=K$:REM GET CALL
470 GOSUB 2220:DT$=DA$:REM GET DATE
480 PRINT "WHAT IS THE STARTING TIME ":GOSUB 2340:UT$=U$:REM GET TIME
490 A$="-":PRINT "WHAT IS THE FREQUENCY";:INPUT A$:FQ$=A$
500 IF LEN (A$)>8 THEN PRINT BL$;E1$;8;E2$:GOTO 490
510 A$="-":PRINT "WHAT IS THE MODE ";:INPUT A$:MD$=A$
520 IF LEN (A$)>3 THEN PRINT BL$;E1$;3;E2$:GOTO 510
530 A$="-":PRINT "WHAT IS THE POWER ";:INPUT A$:PR$=A$
535 IF LEN(A$)>4 THEN PRINT BL$;E1$;4;E2$:GOTO 530
540 A$="-":PRINT "WHAT IS HIS RST ";:INPUT A$:RS$=A$
550 IF LEN (A$)>3 THEN PRINT BL$;E1$;3;E2$:GOTO 540
560 A$="-":PRINT "WHAT RST DID YOU RECEIVE ";:INPUT A$:RR$=A$
570 IF LEN (A$)>3 THEN PRINT BL$;E1$;3;E2$:GOTO 560
590 PRINT "ENTER THE ENDING TIME OF THE QSO ":GOSUB 2340:UE$=U$
600 A$="N":PRINT "IS A QSL BEING SENT (Y/N) ";:INPUT A$:QS$=A$
610 IF LEN (A$)>1 THEN PRINT BL$;E1$;1;E2$:GOTO 600
620 QR$="N"
630 PRINT "ENTER ANY COMMENTS (";C1;" CHARACTERS MAX)"
640 A$="-":INPUT A$:CM$=A$
650 IF LEN(A$)>C1 THEN PRINT BL$;LEN(A$)-C1;" CHARS TOO MANY. RETRY.":GOTO 640
660 LR=LR+1:RN=LR:GOSUB 2760:REM WRITE LOG
670 RN=H:GOSUB 2930:IF P=0 THEN 690
680 RN=P:GOSUB 2930:IF P<>0 THEN 680
690 P=LR+26:GOSUB 2990
700 RN=P:P=0:QL$=KX$:GOSUB 2990
710 RN=0:GOSUB 2590
720 CM$=DA$:DT$=STR$(LR):GOSUB 2760
730 GOTO 360
740 REM FIND A CALL & DISPLAY IT
750 GOSUB 2390:GOSUB 800:REM GET CALL & DISPLAY HEADER
760 RN=H:GOSUB 2930:IF P=0 THEN 790
770 RN=P:GOSUB 2930:IF KX$=QM$ THEN GOSUB 840
780 IF P>0 THEN 770
```

```
790 GOTO 360
800 PRINT:PRINT "RCD#    DATE       TIME     MODE RST  QRT "
810 PRINT "        CALL       FREQ     PWR  S/R  QSL"
820 PRINT "---- ---------- -------- ---- ---  ----"
830 RETURN
840 RN=RN-26:GOSUB 2590:REM READ LOG
850 PRINT RN;TAB(6)DT$;TAB(17)UT$;TAB(26)MD$;TAB(31)RS$;TAB(36)UE$
860 PRINT TAB(6)KL$;TAB(17)FQ$;TAB(26)PR$;TAB(31)RR$;TAB(37)QS$;"/";QR$
870 PRINT TAB(6)"COMMENTS:";CM$
880 PRINT:RETURN
890 GOSUB 800:REM DISPLAY LAST 5 LOG ENTRIES
900 IF LR=0 THEN GOTO 360
910 N=LR-4:IF LR<5 THEN N=1
920 FOR RN=N TO LR:GOSUB 2590:GOSUB 850:NEXT RN
950 GOTO 360
960 REM LIST LOG ON PRINTER
970 N1=0:L2=L1+6
980 PRINT:PRINT "LIST BY:"
990 PRINT "  1. CALL SIGN"
1000 PRINT "  2. PREFIX"
1010 PRINT "  3. DATE"
1020 PRINT "  4. FREQUENCY"
1030 PRINT "  5. MODE"
1035 PRINT "  6. QSLS NOT RECEIVED"
1040 PRINT "  7. COMPLETE LOG"
1050 PRINT:PRINT "WHICH WOULD YOU LIKE ";:INPUT PL
1060 ON PL GOTO 1120,1170,1240,1400,1320,1481,1080
1070 PRINT "INVALID";BL$:GOTO 360
1080 FOR RN=1 TO LR:REM LIST COMPLETE LOG
1090 GOSUB 2590:GOSUB 1630:NEXT RN
1110 GOTO 360
1120 GOSUB 2390:REM LIST BY CALL
1130 RN=H:GOSUB 2930:IF P=0 THEN 360
1140 RN=P:GOSUB 2930:IF KX$=QM$ THEN RN=RN-26:GOSUB 2590:GOSUB 1630
1150 IF P>0 THEN 1140
1160 GOTO 360
1170 PX$="":PRINT "WHAT IS THE PREFIX ";:INPUT PX$:LX=LEN(PX$):REM LIST BY PX
1180 IF PX$="" THEN PRINT BL$;"ERROR":GOTO 360
1190 FOR RN=1 TO LR:GOSUB 2590:G$=KL$:GOSUB 3040
1210 IF LEN(G$)>=LX THEN IF LEFT$ (G$,LX)=PX$ THEN GOSUB 1630
1220 NEXT RN
1230 GOTO 360
1240 PRINT"WHAT IS THE STARTING DATE (MM/DD/YY) ":INPUT DA$:GOSUB 2240
1242 SD$=RIGHT$(DA$,2)+"/"+LEFT$(DA$,5)
1250 PRINT "WHAT IS THE ENDING DATE (MM/DD/YY) ":INPUT DA$:GOSUB 2240
1252 ED$=RIGHT$(DA$,2)+"/"+LEFT$(DA$,5)
1260 IF SD$>ED$ THEN  PRINT BL$;"ERROR":GOTO 360
1270 FOR RN=1 TO LR:GOSUB 2590:X$=RIGHT$(DT$,2)+"/"+LEFT$(DT$,5)
1290 IF X$>=SD$ THEN IF X$<=ED$ THEN GOSUB 1630
1300 NEXT RN
1310 GOTO 360
1320 INPUT "WHAT IS THE MODE ";SM$:REM LIST BY MODE
1330 IF LEN(SM$)>3 THEN PRINT BL$;E1$;3;E2$:GOTO 1320
1340 IF LEN(SM$)<3 THEN FOR J=LEN(SM$)+1 TO 3:SM$=SM$+" ":NEXT J
1350 FOR RN=1 TO LR:GOSUB 2590
1370 IF SM$=MD$ THEN  GOSUB 1630
1380 NEXT RN
1390 GOTO 360
1400 PRINT "WHAT IS THE LOWER FREQUENCY LIMIT ";:INPUT LF:REM LIST BY FREQ
1410 PRINT "WHAT IS THE UPPER FREQUENCY LIMIT ";:INPUT UF
1420 IF UF<LF THEN  PRINT BL$;"ERROR":GOTO 360
1430 FOR RN=1 TO LR:GOSUB 2590:F=VAL(FQ$)
1460 IF F>=LF THEN IF F<=UF THEN GOSUB 1630
1470 NEXT RN
1480 GOTO 360
1481 FOR RN=1 TO LR:REM LIST BY QSL'D
1482 GOSUB 2590
1483 IF QR$="Y" THEN 1485
1484 IF QS$<>"N" THEN IF QS$<>"0" THEN GOSUB 1630
1485 NEXT RN:GOTO 360
```

```
1490 N1=N1+1:REM NEW PAGE
1500 IF L2>=L1+6 THEN 1520
1510 FOR J=L2+1 TO L1+6:PRINT#130,"":NEXT J
1520 PRINT#130,"";SPC(29)"LOG OF ";MK$;SPC(21-LEN(MK$))"PAGE ";N1
1530 PRINT#130,"";SPC(33)"VOL ";VL
1540 PRINT#130,""
1550 PRINT#130,"";SPC(50)"HIS MY       QSL"
1560 PRINT#130,"RCD#   DATE   TIME    CALL      FREQ   MODE";
1561 PRINT#130," PWR  RST RST QRT  S/R";
1570 IF NW<=PW THEN PRINT#130,"     COMMENTS";
1580 PRINT#130,""
1590 PRINT#130,"---- -------- ---- ---------- -------- ---  ----";
1591 PRINT#130," --- --- ---- ---";
1600 IF NW<=PW THEN PRINT#130,"  ---------------";
1610 PRINT#130,""
1620 L2=12:RETURN
1630 IF L2+3>L1-6 THEN GOSUB 1490:REM NEW PAGE
1640 PRINT#130,RN;"   "DT$;SPC(9-LEN(DT$))UT$;SPC(5-LEN(UT$))KL$;
1641 PRINT#130," ";SPC(10-LEN(KL$))FQ$;SPC(9-LEN(FQ$))MD$;SPC(5-LEN(MD$))PR$;
1650 PRINT#130,"";SPC(5-LEN(PR$))RS$;SPC(4-LEN(RS$))RR$;SPC(4-LEN(RR$))UE$;
1651 PRINT#130,"";SPC(5-LEN(UE$))QS$;"/";QR$;
1760 IF NW<=PW THEN PRINT#130,"  ";CM$:L2=L2+1:GOTO 1800
1770 PRINT#130,""
1780 PRINT#130,"      COMMENTS: ";CM$
1790 PRINT#130,"":L2=L2+3
1800 RETURN
1810 PRINT:PRINT "ENTER THE RECORD # TO BE EDITED OR"
1820 A$="":PRINT "<CR> TO EDIT THE LAST LOG ENTRY ";:INPUT A$
1830 IF A$="" THEN RN=LR:GOTO 1850
1840 RN=VAL(A$)
1850 IF RN>LR THEN 2210
1860 IF RN<1 THEN 2210
1870 PRINT C7$:GOSUB 2590:GOSUB 800:GOSUB 850
1880 INPUT "EDIT THIS (Y/N) ";A$
1890 IF A$<>"Y" THEN 360
1900 PRINT "ENTER <CR> TO LEAVE AN ITEM UNCHANGED."
1910 PRINT "ENTER A NEW VALUE TO CHANGE IT."
1920 PRINT:PRINT "DATE: ";DT$;
1922 DA$="":INPUT DA$:IF DA$<>"" THEN GOSUB 2240:DT$=DA$
1930 PRINT "STARTING TIME: ";UT$;
1932 U$="":INPUT U$:IF U$<>"" THEN GOSUB 2350:UT$=U$
1940 PRINT "FREQUENCY: ";FQ$;
1942 A$="":INPUT A$:IF A$="" THEN 1970
1950 IF LEN (A$)>8 THEN PRINT BL$;E1$;8;E2$:GOTO 1940
1960 FQ$=A$
1970 A$="":PRINT "MODE: ";MD$;:INPUT A$:IF A$="" THEN 2000
1980 IF LEN (A$)>3 THEN PRINT BL$;E1$;3;E2$:GOTO 1970
1990 MD$=A$
2000 A$="":PRINT "POWER: ";PR$;:INPUT A$:IF A$="" THEN 2030
2010 IF LEN (A$)>4 THEN PRINT BL$;E1$;4;E2$:GOTO 2000
2020 PR$=A$
2030 A$="":PRINT "RST SENT: ";RS$;:INPUT A$:IF A$="" THEN 2060
2040 IF LEN (A$)>3 THEN PRINT E1$;3;E2$:GOTO 2030
2050 RS$=A$
2060 PRINT "MY RST: ";RR$;:A$="":INPUT A$: IF A$="" THEN 2090
2070 IF LEN (A$)>3 THEN PRINT E1$;3;E2$:GOTO 2060
2080 RR$=A$
2090 U$="":PRINT "ENDING TIME: ";UE$;:INPUT U$:IF U$<>"" THEN GOSUB 2350:UE$=U$
2100 A$="":PRINT "QSL SENT: ";QS$;:INPUT A$:IF A$="" THEN 2130
2110 IF LEN (A$)<>1 THEN PRINT BL$;E1$;1;E2$:GOTO 2100
2120 QS$=A$
2130 A$="":PRINT "QSL RCVD: ";QR$;:INPUT A$:IF A$="" THEN 2160
2140 IF LEN(A$)<>1 THEN PRINT BL$;E1$;1;E2$:GOTO 2130
2150 QR$=A$
2160 A$="":PRINT "COMMENTS: ";CM$:INPUT "          ";A$: IF A$="" THEN 2190
2170 IF LEN (A$)>C1 THEN  PRINT BL$;E1$;C1;E2$:GOTO 2160
2180 CM$=A$
2190 A$="":INPUT "OK (Y/N) ";A$:IF A$<>"Y" THEN 360
2200 GOSUB 2760:GOTO 360
2210 PRINT BL$;"INVALID RECORD #":GOTO 360
```

```
2220 REM GET DATE
2230 PRINT "WHAT IS THE DATE (MM/DD/YY) ";:INPUT DA$
2240 IF LEN(DA$)<6 THEN 2330
2250 IF MID$(DA$,3,1)="/" THEN 2280
2260 IF MID$(DA$,2,1)<>"/" THEN 2330
2270 DA$="0"+DA$
2280 IF MID$(DA$,6,1)="/" THEN 2310
2290 IF MID$(DA$,5,1)<>"/" THEN 2330
2300 DA$=LEFT$(DA$,3)+"0"+RIGHT$(DA$,4)
2310 IF LEN(DA$)<>8 THEN 2330
2320 RETURN
2330 PRINT BL$;"INVALID. RE-ENTER.":GOTO 2220
2340 PRINT "(HHMM) ";:INPUT U$:REM  GET TIME
2350 IF LEN (U$)>4 THEN PRINT BL$;E1$;4;E2$;" RE-ENTER":GOTO 2340
2360 IF U$="" THEN U$="-":RETURN
2370 IF LEN (U$)<4 THEN FOR J=LEN(U$)+1 TO 4:U$="0"+U$:NEXT J
2380 RETURN
2390 REM GET CALL SIGN
2400 PRINT "WHAT IS THE CALL SIGN ":INPUT K$
2410 IF LEN(K$)>10 THEN PRINT BL$;E1$;10;E2$:GOTO 2400
2420 IF LEN(K$)<4 THEN 2460
2422 FOR J=1 TO LEN(K$):IF MID$(K$,J,1)="/" THEN J=J-1:GOTO 2425
2423 NEXT J
2425 IF J>6 THEN J=6
2426 KX$=MID$(K$,1,J)
2430 H=ASC(MID$(K$,4,1))-ASC("A")+1
2440 IF H<1 THEN 2460
2450 IF H<=26 THEN RETURN
2460 PRINT BL$;"INVALID CALL":GOTO 2400
2470 REM MENU
2475 PRINT:PRINT:PRINT " ";HI$;"                                              "
2480 PRINT " ";HI$;"                LOG BOOK MENU                "
2490 PRINT " ";HI$;"                                             "
2500 PRINT " ";HI$;" '?' SHOWS THIS MENU                         "
2510 PRINT " ";HI$;" 'QSO' ENTERS A NEW QSO INTO THE LOG         "
2520 PRINT " ";HI$;" 'QRT' EXITS THE LOG PROGRAM                 "
2530 PRINT " ";HI$;" 'LIST' PRINTS LOG ON THE PRINTER            "
2540 PRINT " ";HI$;" 'FIND' FINDS A CALL IN THE LOG BOOK         "
2550 PRINT " ";HI$;" 'LAST' SHOWS LAST 5 LOG ENTRIES             "
2560 PRINT " ";HI$;" 'EDIT' EDITS A LOG ENTRY                    "
2570 PRINT " ";HI$;"                                             "
2580 PRINT LO$:PRINT:RETURN
2590 REM READ LOG
2600 R1=RN+1:GOSUB 7000
2610 R9=1:GOSUB 7100:INPUT#3,DT$
2620 R9=10:GOSUB 7100:INPUT#3,UT$
2630 R9=15:GOSUB 7100:INPUT#3,KL$
2640 R9=26:GOSUB 7100:INPUT#3,RS$
2650 R9=30:GOSUB 7100:INPUT#3,RR$
2660 R9=34:GOSUB 7100:INPUT#3,FQ$
2670 R9=43:GOSUB 7100:INPUT#3,MD$
2680 R9=47:GOSUB 7100:INPUT#3,PR$
2690 R9=52:GOSUB 7100:INPUT#3,UE$
2700 R9=57:GOSUB 7100:INPUT#3,QS$
2710 R9=59:GOSUB 7100:INPUT#3,QR$
2750 R9=61:GOSUB 7100:INPUT#3,CM$:GOSUB 6000:RETURN
2760 REM WRITE LOG
2780 R1=RN+1:GOSUB 7000
2790 R9=1:GOSUB 7100:PRINT#3,DT$
2800 R9=10:GOSUB 7100:PRINT#3,UT$
2810 R9=15:GOSUB 7100:PRINT#3,KL$
2820 R9=26:GOSUB 7100:PRINT#3,RS$
2830 R9=30:GOSUB 7100:PRINT#3,RR$
2840 R9=34:GOSUB 7100:PRINT#3,FQ$
2850 R9=43:GOSUB 7100:PRINT#3,MD$
2860 R9=47:GOSUB 7100:PRINT#3,PR$
2870 R9=52:GOSUB 7100:PRINT#3,UE$
2880 R9=57:GOSUB 7100:PRINT#3,QS$
2890 R9=59:GOSUB 7100:PRINT#3,QR$
2920 R9=61:GOSUB 7100:PRINT#3,CM$:GOSUB 6000:RETURN
```

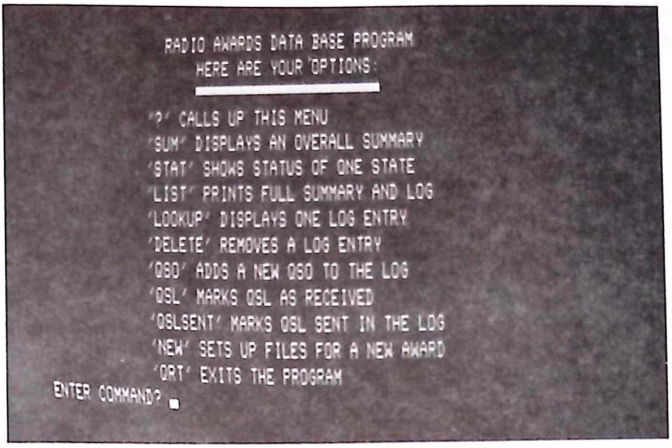
```
2930 REM READ F2
2940 QL$=QT$(RN):P=PT(RN)
2960 G$=QL$:QM$=G$
2980 RETURN
2990 REM WRITE F2
3000 QT$(RN)=QL$:PT(RN)=P:RETURN
3040 REM REMOVE TRAILING BLANKS FROM G$
3050 FOR J=1 TO LEN(G$):IF MID$(G$,J,1)=" " THEN 3070
3060 NEXT J
3070 IF J=1 THEN G$="":RETURN
3080 G$=MID$(G$,1,J-1)
3090 RETURN
3100 PRINT:PRINT:PRINT TAB(13)" ";HI$;"  WARNING  "
3101 PRINT:PRINT "IF YOU CREATE A NEW LOG, ANY EXISTING"
3102 PRINT "LOG DATA FILES ON THIS DISK WILL BE"
3103 PRINT "DESTROYED.  DO YOU STILL WANT TO SET UP"
3104 A$="":PRINT "NEW LOG FILES (Y/N)";:INPUT A$
3105 IF A$<>"Y" THEN 205
3109 PRINT:PRINT TAB(8)"INITIALIZING LOG FILES"
3110 PRINT#15,"S"+F1$
3112 OPEN3,8,3,F1$+",L,"+CHR$(C2)
3120 PRINT:PRINT "WHAT IS YOUR CALL ";:INPUT MK$
3130 PRINT:PRINT "WHAT VOLUME NUMBER IS THIS LOG ";:INPUT VL
3140 PRINT:GOSUB 2220:CD$=DA$
3150 DT$="0":LR=0
3160 KL$=MK$:FQ$=CD$
3170 UT$=STR$(VL):CM$=CD$
3175 RS$="X":RR$="X":MD$="X":PR$="X"
3176 UE$="X":QS$="X":QR$="X"
3180 RN=0:GOSUB 2760
3190 P=0:QL$="!"
3200 FOR RN=1 TO 26:GOSUB 2990:NEXT RN
3222 GOSUB 3500
3230 GOTO 230
3500 PRINT#15,"S"+F2$
3510 OPEN4,8,4,F2$+",S,W":GOSUB 6000
3520 FOR J=1 TO LR+26:PRINT#4,QT$(J):PRINT#4,PT(J):NEXT J
3540 GOSUB 6000:CLOSE4:RETURN
6000 INPUT#15,EA,EB$,EC,ED:IF EA<20 THEN RETURN:REM READ ERROR CHANNEL
6010 IF EA=50 THEN RETURN
6020 IF EA=63 THEN RETURN
6030 PRINT EA;EB$;EC;ED:STOP:RETURN
7000 RH=INT(R1/256):RL=R1-256*RH:RETURN
7100 PRINT#15,"P"CHR$(3)CHR$(RL)CHR$(RH)CHR$(R9):GOTO 6000
```

THE RADIO AWARDS DATA BASE PROGRAM

What it does

The AWARDS program will set up and manage a data base for almost any amateur radio operating award, such as Five-Band DXCC, Worked All States, Worked All Zones, VHF-UHF Century Club, and many others.

The key to the program's versatility is an initial data file called AWRDINFO (there's only so much you can say in eight characters, which is the limit for filenames on many systems). This file, which you set up for each awards program you're interested in, includes a few details about the particular award, such as its name, the number of bands, and then the names and abbreviations for the geographic or other units on which the award is based. The AWARDS program reads in the AWRDINFO file and sets up two other data files to keep records as you work toward the award.


RADIO AWARDS DATA BASE PROGRAM
HERE ARE YOUR OPTIONS:
'?' CALLS UP THIS MENU
'SUM' DISPLAYS AN OVERALL SUMMARY
'STAT' SHOWS STATUS OF ONE STATE
'LIST' PRINTS FULL SUMMARY AND LOG
'LOOKUP' DISPLAYS ONE LOG ENTRY
'DELETE' REMOVES A LOG ENTRY
'QSO' ADDS A NEW QSO TO THE LOG
'QSL' MARKS QSL AS RECEIVED
'QSLSENT' MARKS QSL SENT IN THE LOG
'NEW' SETS UP FILES FOR A NEW AWARD
'QRT' EXITS THE PROGRAM
ENTER COMMAND?

The AWARDS DATA BASE program offers many features, as the main menu shows. Depending on the particular award for which it is customized, it will display your country, state, zone, or grid locator totals, with a breakdown band by band. The program keeps track of QSLs sent and received, too.

The AWARDS program tells you how you're doing in working toward the award, either on an overall basis or band by band—or both. It displays the list of stations you've worked in any or all geographic units, indicating which ones have and have not sent QSL cards. The program also prints a hard copy of most of this information, including a list of all individual entries and a full summary of where you stand. You can generate these status reports as often as you like.

The program will handle enough data to keep track of almost any awards program. On an Apple disk, you can store some 4000 records, covering at least 500 different geographic entities. If your goal is five-band DXCC, you can maintain separate records for all recognized DXCC countries on each band. For awards with smaller data bases (such as WAS or WAZ), it's a cinch.

Setup instructions

To get started with the AWARDS DATA BASE, there are only a few things to do other than typing the program as listed for your particular system. The information stored for each QSO is so brief (just 34 characters) that there is no problem fitting each entry on a single line with this program; there is no need to modify the print width for your printer as in the LOGBOOK. If you have any standard 80-column printer, everything including the summaries will fit nicely.

The major thing you have to do when you set up the program for a particular award is to create the AWRDINFO data file, using any text editor. If you prefer, you can use the special program we have included in this chapter to set up this file. There are separate listings in this short program for Apple, Commodore and TRS-80 (or CP/M).

The adjacent figures show samples of what an AWRDINFO file should look like. Omit the explanatory comments alongside the entries in the Five-Band WAS file, of course. In creating entries for other awards,

```
FIVE BAND WAS       (name of award)
STATE               (name of unit on which award is based)
5                   (number of bands)
50                  (number of states)
80                  (names of the bands)
40
20
15
10
AL,ALABAMA          (short designation and full name for each state)
AK,ALASKA
AZ,ARIZONA
AR,ARKANSAS
CA,CALIFORNIA
CO,COLORADO
CT,CONNECTICUT
DE,DELAWARE
FL,FLORIDA
GA,GEORGIA
HI,HAWAII
ID,IDAHO
IL,ILLINOIS
IN,INDIANA
IA,IOWA
KS,KANSAS
KY,KENTUCKY
LA,LOUISIANA
ME,MAINE
MD,MARYLAND
MA,MASSACHUSETTS
MI,MICHIGAN
MN,MINNESOTA
MS,MISSISSIPPI
MO,MISSOURI
MT,MONTANA
NE,NEBRASKA
NV,NEVADA
NH,NEW HAMPSHIRE
NJ,NEW JERSEY
NM,NEW MEXICO
NY,NEW YORK
NC,N. CAROLINA
ND,N. DAKOTA
OH,OHIO
OK,OKLAHOMA
OR,OREGON
PA,PENNSYLVANIA
RI,RHODE ISL
SC,S. CAROLINA
SD,S.DAKOTA
TN,TENNESSEE
TX,TEXAS
UT,UTAH
VT,VERMONT
VA,VIRGINIA
WA,WASHINGTON
WV,W.VIRGINIA
WI,WISCONSIN
WY,WYOMING
```

This is a sample AWRDINFO data file for use with the AWARDS DATA BASE program. This one is for the five-band Worked All States (WAS) award.

```
FIVE BAND DXCC
COUNTRY
5
318
80
40
20
15
10
A2,BOTSWANA
A3,TONGA IS
A4X,OMAN
A5,BHUTAN
A6X,UNITED ARAB EMIRATES
A7X,QATAR
A9X,BAHRAIN
AP,PAKISTAN
BV,TAIWAN
BY,CHINA
C2,NAURU
C3,ANDORRA
C5,THE GAMBIA
C6,BAHAMAS
C9,MOZAMBIQUE
CE,CHILE
CE0A,EASTER IS
CE0X,SAN FELIX
CE0Z,JUAN FERNANDEZ
CO,CUBA
CN,MOROCCO
CP,BOLIVIA
CR9,MACAO
CT,PORTUGAL
CT2,AZORES IS
CT3,MADEIRA IS
CX,URUGUAY
D2,ANGOLA
D4,CAPE VERDE IS
D6,COMOROS IS
DJ,WEST GERMANY
DU,PHILIPPINES
EA,SPAIN
—(and so on, through countries list)—
```

This sample of an AWRDINFO file is for Five-Band DXCC. Note that each country is identified by a single prefix, even if there are other prefixes used by amateurs there. Prefixes are limited to four characters

```
VHF-UHF CENTURY CLUB
GRID BLOCK
7
150
50
144
220
432
1296
        (continued)
```

This AWRDINFO file is set up for VHF-UHF Century Club for a station in California. The four-character grid locators are determined by latitude and longitude; the names were chosen arbitrarily to describe the places.

```
2304
10GC
DM22,YUMA-EL CENTRO
DM12,SAN DIEGO
DM23,BLYTHE
DM13,INDIO-ANAHEIM
DM03,L.A.SOUTH
CM93,LOTSA LUCK
DM24,NEEDLES
DM14,HIGH DESERT
DM04,L.A.-SANTA BARBARA
CM94,LOMPOC
DM25,MT.POTOSI
DM15,RIDGECREST
DM05,BAKERSFIELD
CM95,SAN LUIS OBISPO
DM26,LAS VEGAS
DM16,DEATH VALLEY
DM06,FRESNO S.E.
CM96,SALINAS-COALINGA
CM86,SANTA CRUZ
DM07,YOSEMITE SOUTH
CM97,SAN JOSE-SONORA
CM87,THE CITY
DM08,E.SIERRA
CM98,SACRAMENTO
CM88,SANTA ROSA
DM09,RENO EAST
CM99,CHICO EAST
CM89,FT. BRAGG EAST
—(and so on, through workable grid blocks)—
```

follow this format. Be sure to put each item of data on a separate line, except for the short designations and place names, which go on a single line for each place.

Operating notes

Once you have a suitable AWRDINFO file (that's the name you must give your information file unless you change the name the AWARDS DATA BASE program expects), begin by running the program with a new disk in your first disk drive. The advice given regarding LOGBOOK—to use a disk that doesn't have a lot of other programs on it—applies here too, since this program also requires considerable disk space as the data base grows.

When you start with a new disk, the program will advise you to select the "NEW" option on the menu. The other two data files will then be set up by your computer (you don't have to do anything yourself to make it happen). But be sure you really want to create new files when you type NEW. If you've already used the program to log some contacts, you'll lose them if you use the NEW command. You'll see a warning about this after you type NEW but before the new files are actually set up.

Once you have the set of data files, be sure the disk containing these files is on your logged disk drive whenever you run the program.

When the files are ready and you're set to begin using the AWARDS program, the menu will summarize the options you have. The QSO command is the one most often used; it adds each new QSO to the data base. A good way to use the program is to run it periodically to enter new countries (or states or whatever the category is), and then run the LIST command to print out a full summary of where you stand on all bands. If the computer is near your operating position, of course, you can leave the program up and ready to go whenever you're chasing those elusive new ones. That allows you to check where you stand regarding any place you hear that sounds interesting (using the STAT command).

However, you may not want the computer to search through a bulging data file before you call a rare one. As is true of the LOGBOOK program, the data search and retrieval functions take time, particularly on certain computers. It's a good idea to keep a recent printout handy to let you know where you stand. And if there's any doubt, work the DX first and ask questions later!

An accompanying photo shows the main menu of the AWARDS DATA BASE program, displaying the various command options.

Modifications for CP/M

The TRS-80 and CP/M listing of the AWARDS program requires only a few changes for use on a CP/M system running Microsoft BASIC (MBASIC.COM). They are:

1. Delete the CLEAR statement in line 40;
2. Add your own video highlighting commands to line 40;
3. Delete all TRS-80 clear-screen commands (CLS) and replace them with your system's comparable command. The CLS command is in lines 40, 370, and 3340.
4. You may wish to increase the tabbing to center the menus and other display features if you have a 24 × 80 video display. Also, line 3650 is there only to prevent scroll-off on a 16-line display. Those with 24-line displays can delete that, too.

AWARDS DATA BASE (TRS-80 AND CP/M)

```
10 REM RADIO AWARDS DATA BASE PROGRAM (TRS-80)
20 REM FOR MULTIBAND DXCC, WAS, WAZ, VUCC AND OTHER AWARDS.
30 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO., INC.
40 CLEAR 1000:CLS:HI$="":LO$="":BL$=""
50 REM HIGHLIGHTING CODES (FOR CP/M SYSTEMS)
70 PRINT:PRINT
100 F1$="AWRDINFO"
110 F2$="AWRDBAND"
120 F3$="AWRDCALL":R3=34
150 PRINT:PRINT
160 PRINT TAB(19)HI$;"DATA BASE FOR RADIO AWARDS";LO$
165 PRINT TAB(19)HI$;STRING$(26,42);LO$
170 PRINT: PRINT:PRINT
200 OPEN "I",1,F1$
220 INPUT#1,AN$:INPUT#1,AM$
230 INPUT#1,NB:INPUT#1,NA
240 DIM BN$(NB),P$(NA),PX$(NA)
250 DIM BS(NB),RP(NB),X(2,NB)
260 R2=4
270 FOR J=1 TO NB:INPUT#1,BN$(J)
271 IF LEN(BN$(J))>4 THEN PRINT BL$;"TOO MANY CHARACTERS IN ";BN$(J);". MAX IS 4
    "
272 NEXT J
```

```
280 FOR J=1 TO NA
290 INPUT#1,P$(J),PX$(J)
292 IF LEN(P$(J))>5 THEN PRINT BL$;"TOO MANY CHARACTERS IN ";P$(J);". MAX IS 5"
300 NEXT J:CLOSE
312 OPEN "R",2,F2$,R2
313 FIELD 2,2 AS BS$,2 AS RP$
314 OPEN "R",3,F3$,R3
315 FIELD 3,6 AS KL$,6 AS DT$,4 AS UT$,3 AS RS$,3 AS RR$,3 AS MD$,6 AS QS$,1 AS
    QR$,2 AS NX$
318 IF LOF(3)=0 THEN 3610
320 RN=0:GOSUB 2690:LH=VAL(KL$)
340 PRINT:PRINT TAB(32-LEN(AN$)/2)AN$
350 GOSUB 3110:REM MENU
360 PRINT "ENTER COMMAND";:C$="":INPUT C$
365 PRINT
370 IF C$="?" THEN CLS:GOSUB 3110:GOTO 360
380 IF C$="SUM" THEN GOSUB 490:GOTO 360
390 IF C$="STAT" THEN GOSUB 1580:GOTO 360
400 IF C$="LIST" THEN GOSUB 1780:GOTO 360
410 IF C$="LOOKUP" THEN GOSUB 1480:GOTO 360
420 IF C$="QSL" THEN GOSUB 1110:GOTO 360
430 IF C$="DELETE" THEN GOSUB 1290:GOTO 360
440 IF C$="QSO" THEN GOSUB 780:GOTO 360
450 IF C$="QSLSENT" THEN GOSUB 2300:GOTO 360
460 IF C$="NEW" THEN GOSUB 3340:GOTO 360
470 IF C$="QRT" THEN CLOSE:END
480 PRINT:PRINT BL$;"INVALID COMMAND. PLEASE TRY AGAIN":GOTO 360
490 PRINT:REM SUMMARY
500 PRINT:PRINT "===== OVERALL SUMMARY ====="
510 PRINT:PRINT TAB(6)"NUMBER OF ";
515 IF RIGHT$(AM$,1)<>"Y" THEN PRINT AM$;"S":GOTO 520
516  PRINT LEFT$(AM$,LEN(AM$)-1);"IES"
520 PRINT "BAND  WORKED  CONFMD  UNCONFMD"
530 PRINT "----  ----  ----  ----"
540 GOSUB 4800
660 FOR J=1 TO NB
670 PRINT BN$(J);TAB(8)X(1,J);TAB(16)X(2,J);TAB(25)NA-X(2,J)
710 NEXT J
720 PRINT "      ----  ----  ----"
730 PRINT "ALL   ";Y1;TAB(16)Y2;TAB(25)NA-Y2
760 PRINT:PRINT "TOTAL LOGGED CONTACTS=";LH
770 RETURN
780 REM ADD A CONTACT TO LOG
790 GOSUB 2820:GOSUB 2980
800 IF BS(B)<2 THEN 840
810 PRINT AM$;" ALREADY CONFIRMED!";BL$
820 A$="":INPUT "DO YOU WANT TO ENTER A NEW QSO ANYWAY (Y/N)";A$
830 IF A$<>"Y" THEN RETURN
840 Z$(1)="":INPUT "WHAT IS THE CALL";Z$(1):IF LEN(Z$(1))>6 THEN 840
842 IF LEN(Z$(1))<3 THEN 840
850 GOSUB 4000
860 IF BS(B)=0 THEN 910
870 RN=RP(B)
880 GOSUB 2690
890 IF KL$=Z$(1) THEN PRINT "DUPLICATE CONTACT";BL$:GOSUB 3260:GOSUB 3300
900 IF NX>0 THEN RN=NX:GOTO 880
910 D$="":INPUT "WHAT IS THE DATE OF THE QSO (MM/DD/YY)";D$
920 GOSUB 2870:Z$(2)=DA$
930 Z$(3)="":INPUT "WHAT IS THE UTC OF THE QSO (HHMM)";Z$(3)
940 IF LEN(Z$(3))<>4 THEN PRINT BL$;"UTC MUST BE 4 DIGITS. RE-ENTER.":GOTO 930
950 Z$(4)="":INPUT "WHAT IS THE RST SENT";Z$(4)
960 IF LEN(Z$(4))>3 THEN PRINT BL$;"ERROR. RE-ENTER":GOTO 950
970 Z$(5)="":INPUT "WHAT WAS THE RST RECEIVED";Z$(5)
980 IF LEN(Z$(5))>3 THEN PRINT BL$;"ERROR. RE-ENTER":GOTO 970
990 Z$(6)="":INPUT "WHAT WAS THE MODE";Z$(6)
1000 IF LEN(Z$(6))>3 THEN Z$(6)=LEFT$(Z$(6),3)
1010 A$="":INPUT "IS A QSL BEING SENT (Y/N)";A$
1020 Z$(7)="      ":IF A$="Y"  THEN Z$(7)=Z$(2)
1030 Z$(8)=" ":Z=0
1040 LH=LH+1:IF BS(B)>0 THEN 1080
```

```
1050 BS(B)=1:RP(B)=LH
1060 GOSUB 2490
1070 RN=LH:GOSUB 2660:GOTO 1100
1080 NX=LH:GOSUB 2550
1090 RN=LH:GOSUB 2660
1100 RN=0:GOSUB 3540:RETURN
1110 REM CONFIRM A CONTACT
1120 GOSUB 2820:GOSUB 2980:NX=RP(B):S3=RN
1130 Z$(1)="":INPUT "WHAT IS THE CALL TO BE CONFIRMED ";Z$(1)
1140 GOSUB 4000
1150 S4=0
1160 IF NX>0 THEN 1190
1170 IF S4=1 THEN RETURN
1180 PRINT "CALL NOT FOUND";BL$:RETURN
1190 RN=NX:GOSUB 2690
1200 IF Z$(1)<>KL$ THEN 1160
1210 S4=1:GOSUB 3260:GOSUB 3300
1220 PRINT:A$="":INPUT "CONFIRM THIS QSO (Y/N) ";A$
1230 IF A$<>"Y" THEN 1160
1240 LSET QR$="Y":GOSUB 2550
1250 BS(B)=2:RN=S3:GOSUB 2490
1260 PRINT:PRINT TAB(15)"CONFIRMED"
1270 IF QS$="       " THEN PRINT:PRINT BL$;"YOU NEED TO SEND A QSL!"
1280 RETURN
1290 REM DELETE A CONTACT
1300 GOSUB 2820:GOSUB 2980:NX=RP(B):S1=0:S2=0:S3=RN
1310 Z$(1)="":INPUT "WHAT IS THE CALL TO BE DELETED ";Z$(1)
1320 GOSUB 4000
1330 S4=0
1340 IF NX=0 THEN 1170
1350 RN=NX:GOSUB 2690:IF QR$="Y" THEN S2=S2+1
1360 IF LEFT$(KL$,1)<>"*" THEN S1=S1+1
1370 IF Z$(1)<>KL$ THEN 1340
1380 PRINT:PRINT KL$;" ";DT$;" ";UT$;" ";BN$(B);" ";MD$;" ";QR$
1390 PRINT:A$="":INPUT "DELETE THIS QSO (Y/N) ";A$
1400 IF A$<>"Y" THEN S4=1:GOTO 1340
1410 IF QR$="Y" THEN S2=S2-1
1420 LSET KL$="*      ":LSET QR$=" ":GOSUB 2550
1430 IF NX=0 THEN 1460
1440 RN=NX:GOSUB 2690:IF QR$="Y" THEN S2=S2+1
1450 IF LEFT$(KL$,1)<>"*" THEN S1=S1+1
1460 BS(B)=0:IF S1>1 THEN BS(B)=1:IF S2>0 THEN BS(B)=2
1470 RN=S3:GOSUB 2490:RETURN
1480 REM LIST A CALL
1490 GOSUB 2820:GOSUB 2980:NX=RP(B)
1500 PRINT:Z$(1)="":INPUT "WHAT IS THE CALL ";Z$(1)
1510 GOSUB 4000
1520 S4=0
1530 IF NX=0 THEN 1170
1540 RN=NX:GOSUB 2690
1550 IF Z$(1)<>KL$ THEN 1530
1560 IF S4=0 THEN GOSUB 3260
1570 GOSUB 3300:S4=1:GOTO 1530
1580 REM LIST A CATEGORY
1590 GOSUB 2980
1600 PRINT:PRINT "BAND  STATUS"
1610 PRINT "----  ------"
1620 FOR J=1 TO NB
1630 PRINT BN$(J);TAB(7)"";:IF BS(J)=2 THEN PRINT "CONFIRMED"
1640 IF BS(J)=1 THEN PRINT "WORKED"
1650 IF BS(J)=0 THEN PRINT "NEED"
1660 NEXT J
1670 PRINT:A$="":INPUT "WANT A DISPLAY OF LOG ENTRIES (Y/N) ";A$:IF A$<>"Y" THEN
     RETURN
1680 FOR J=1 TO NB:NX=RP(J)
1690 IF BS(J)=0 THEN 1760
1700 PRINT:PRINT "CAT & BAND= ";C$;", ";BN$(J);
1710 IF BS(J)=1 THEN PRINT "   NOT CONFIRMED":GOTO 1730
1720 PRINT "   CONFIRMED"
1730 GOSUB 3260
```

```
1740 IF NX>0 THEN RN=NX:GOSUB 2690:GOSUB 3300:GOTO 1740
1750 IF J<NB THEN PRINT:A$="":INPUT "<ENTER> TO CONTINUE";A$
1760 NEXT J
1770 RETURN
1780 REM LIST ALL CATEGORIES TO PRINTER
1790 LPRINT:LPRINT "      ***  ";AN$;"  ***"
1800 LPRINT:GOSUB 4400:LPRINT:LPRINT
1810 LPRINT:K=INT(NB*5/2)-1
1820 FOR J=1 TO K:LPRINT "=";:NEXT J:LPRINT " SUMMARY BY ";AM$;" ";
1830 FOR J=1 TO K:LPRINT "=";:NEXT J:LPRINT:LPRINT
1840 LPRINT " - = NOT WORKED;  W = WORKED & UNCONFMD;  C = CONFIRMED":LPRINT
1850 LPRINT AM$;"   BAND"
1870 FOR J=1 TO NB
1880 LPRINT TAB(J*5+2)BN$(J);
1890 NEXT:LPRINT " ALL  ";AM$
1900 LPRINT "-----  ";:FOR J=1 TO NB+1:LPRINT "--   ";:NEXT J:LPRINT "----------
     "
1910 FOR JJ=1 TO NA
1920 RN=JJ:GOSUB 2440:REM GET F2 REC
1930 S1=0
1940 LPRINT P$(JJ);TAB(7)": ";
1950 FOR J=1 TO NB
1960 IF BS(J)=0 THEN LPRINT "-    ";
1970 IF BS(J)=1 THEN LPRINT "W    ";:IF S1=0 THEN S1=1
1980 IF BS(J)=2 THEN LPRINT "C    ";:S1=2
1990 NEXT J
2000 IF S1=0 THEN LPRINT "-   ";PX$(JJ)
2010 IF S1=1 THEN LPRINT "W   ";PX$(JJ)
2020 IF S1=2 THEN LPRINT "C   ";PX$(JJ)
2030 NEXT JJ
2040 LPRINT:LPRINT:LPRINT "============================ LOG ====================
     ========":LPRINT
2050 LPRINT AM$;TAB(10)"BAND  CALL    DATE    UTC SNT RCD MODE QSL-SENT  CFMD"
2070 LPRINT "-------- ---- ------ -------- ---- --- --- ---- --------  -----"
2080 LPRINT
2090 FOR JJ=1 TO NA:RN=JJ:GOSUB 2440
2100 FOR KK=1 TO NB
2110 IF RP(KK)>0 THEN RN=RP(KK):GOSUB 2140
2120 NEXT KK:NEXT JJ
2130 LPRINT:LPRINT:RETURN
2140 IF RN=0 THEN RETURN
2150 GOSUB 2690:IF LEFT$(KL$,1)<>"*" THEN GOSUB 2170
2160 RN=NX:GOTO 2140
2170 LPRINT P$(JJ);TAB(10)BN$(KK);TAB(15)KL$;
2180 LPRINT TAB(22)LEFT$(DT$,2);"/";MID$(DT$,2,2);"/";RIGHT$(DT$,2);
2190 LPRINT TAB(31)UT$;TAB(36)RS$;TAB(40)RR$;TAB(44)MD$;
2200 IF QS$<>"       " THEN LPRINT TAB(49)LEFT$(QS$,2);"/";MID$(QS$,2,2);"/";RIGH
     T$(QS$,2);
2210 LPRINT TAB(59)QR$
2290 RETURN
2300 REM MARK QSL SENT
2310 GOSUB 2820:GOSUB 2980:NX=RP(B)
2320 Z$(1)="":INPUT "WHAT IS THE CALL TO BE QSL'D ";Z$(1)
2330 GOSUB 4000
2340 S4=0
2350 IF NX=0 THEN 1170
2360 RN=NX:GOSUB 2690
2370 IF Z$(1)<>KL$ THEN 2350
2380 GOSUB 3260:GOSUB 3300
2390 PRINT:A$="":INPUT "QSL THIS QSO (Y/N) ";A$
2400 IF A$<>"Y" THEN S4=1:GOTO 2350
2410 PRINT:D$="":INPUT "WHAT IS THE DATE (MM/DD/YY) ";D$
2420 GOSUB 2870:LSET QS$=DA$
2430 GOSUB 2550:RETURN
2440 FOR JK=1 TO NB:RO=(RN-1)*NB+JK
2450 GET 2,RO:BS(JK)=CVI(BS$):RP(JK)=CVI(RP$)
2460 NEXT JK
2480 RETURN
2490 REM WRITE FILE 2
2500 FOR JK=1 TO NB:RO=(RN-1)*NB+JK
```

```
2510 LSET BS$=MKI$(BS(JK))
2520 LSET RP$=MKI$(RP(JK))
2530 PUT 2,RO
2535 NEXT JK
2540 RETURN
2550 REM WRITE LOG ENTRY
2630 LSET NX$=MKI$(NX)
2640 PUT 3,RN+1
2650 RETURN
2660 LSET KL$=Z$(1):LSET DT$=Z$(2):LSET UT$=Z$(3):LSET RS$=Z$(4)
2670 LSET RR$=Z$(5):LSET MD$=Z$(6):LSET QS$=Z$(7):LSET QR$=Z$(8)
2680 NX=Z:GOTO 2550
2690 GET 3,RN+1
2700 NX=CVI(NX$)
2810 RETURN
2820 IF NB=1 THEN B=1:RETURN
2822 PRINT:B$="":INPUT "WHAT BAND ";B$
2830 FOR B=1 TO NB:IF B$=BN$(B) THEN RETURN
2840 NEXT B
2850 PRINT BL$;"INVALID BAND. VALID BANDS ARE:"
2860 FOR J=1 TO NB:PRINT TAB(5)BN$(J):NEXT J:GOTO 2820
2870 REM CHECK & COMPRESS DATE
2880 IF LEN(D$)>8 THEN 2970
2890 IF LEN(D$)<6 THEN 2970
2900 IF MID$(D$,2,1)="/" THEN D$="0"+D$
2910 IF MID$(D$,3,1)<>"/" THEN 2970
2920 DA$=LEFT$(D$,2)
2930 IF MID$(D$,5,1)="/" THEN DA$=DA$+"0"+MID$(D$,4,1):GOTO 2960
2940 IF MID$(D$,6,1)<>"/" THEN 2970
2950 DA$=DA$+MID$(D$,4,2)
2960 DA$=DA$+RIGHT$(D$,2):RETURN
2970 D$="":INPUT "INVALID DATE. RE-ENTER (MM/DD/YY) ";D$:GOTO 2880
2980 REM GET CATEGORY
2990 PRINT:PRINT "WHAT IS THE ";AM$;:C$="":INPUT " ";C$
3000 FOR J=1 TO NA
3010 IF C$=P$(J) THEN PRINT AM$;" IS ";PX$(J):GOTO 3090
3020 NEXT J
3030 PRINT BL$;"INVALID ";AM$;". POSSIBLE VALUES ARE:"
3040 IF LEN(C$)<1 THEN 2990
3050 AA$=LEFT$(C$,1)
3060 FOR J=1 TO NA
3070 IF AA$=LEFT$(P$(J),1) THEN PRINT TAB(5)P$(J);TAB(15)PX$(J)
3080 NEXT J:GOTO 2990
3090 RN=J:GOSUB 2440:REM GET BAND REC FROM F2
3100 RETURN
3110 PRINT
3111 PRINT TAB(12)"   RADIO AWARDS DATA BASE PROGRAM"
3112 PRINT TAB(12)"        HERE ARE YOUR OPTIONS:"
3113 PRINT TAB(19)STRING$(22,42)
3120 PRINT TAB(12)" '?' CALLS UP THIS MENU"
3130 PRINT TAB(12)" 'SUM' DISPLAYS AN OVERALL SUMMARY"
3140 PRINT TAB(12)" 'STAT' SHOWS STATUS OF ONE ";AM$
3160 PRINT TAB(12)" 'LIST' PRINTS FULL SUMMARY AND LOG"
3170 PRINT TAB(12)" 'LOOKUP' DISPLAYS ONE LOG ENTRY"
3180 PRINT TAB(12)" 'DELETE' REMOVES A LOG ENTRY"
3190 PRINT TAB(12)" 'QSO' ADDS A NEW QSO TO THE LOG"
3200 PRINT TAB(12)" 'QSL' MARKS QSL AS RECEIVED"
3210 PRINT TAB(12)" 'QSLSENT' MARKS QSL SENT IN THE LOG"
3220 PRINT TAB(12)" 'NEW' SETS UP FILES FOR A NEW AWARD"
3230 PRINT TAB(12)" 'QRT' EXITS THE PROGRAM"
3250 RETURN
3260 PRINT:REM DISPLAY A CALL
3270 PRINT " CALL    DATE    UTC SNT RCD MDE QSLSNT C"
3280 PRINT "------ ------ ---- --- --- --- ------ -"
3290 RETURN
3300 IF LEFT$(KL$,1)="*" THEN RETURN
3310 PRINT KL$;TAB(8)DT$;TAB(15)UT$;TAB(20)RS$;
3320 PRINT TAB(24)RR$;TAB(28)MD$;TAB(32)QS$;TAB(39)QR$
3330 RETURN
3340 CLS:PRINT:REM CREATE F2 AND F3 FILES
```

```
 3350 PRINT TAB(15)"WARNING"
 3355 PRINT TAB(15)STRING$(7,42)
 3360 PRINT "THIS WILL ERASE EXISTING LOG FILES AND"
 3370 PRINT "LOGGED CONTACTS TO CREATE A NEW DATA"
 3380 PRINT "BASE. THIS COMMAND USES A MASTER FILE:"
 3390 PRINT "'AWRDINFO' (WHICH MUST BE GENERATED BY A"
 3400 PRINT "WORD PROCESSOR OR TEXT EDITOR PROGRAM)."
 3410 PRINT:A$="":INPUT "DO YOU STILL WANT TO CREATE NEW FILES  (Y/N) ";A$
 3420 IF A$<>"Y" THEN RETURN
 3422 PRINT:PRINT TAB(15)"...WORKING"
 3430 FOR J=1 TO NB:BS(J)=0:RP(J)=0:NEXT J
 3460 FOR RN=1 TO NA
 3470 GOSUB 2500
 3480 NEXT RN
 3520 LH=0:RN=0:NX=0
 3530 GOSUB 3540:PRINT:PRINT TAB(17)"-DONE-":GOSUB 3110:RETURN
 3540 FOR J=2 TO 8:Z$(J)="":NEXT J
 3550 Z$(1)=STR$(LH):GOSUB 2660
 3600 RETURN
 3610 REM NO F3 FILE FOUND
 3620 PRINT TAB(28)"WARNING"
 3630 PRINT TAB(13)"DATA FILES ARE MISSING. USE THE 'NEW'"
 3640 PRINT TAB(13)"COMMAND BEFORE PROCEEDING"
 3650 PRINT:XX$="":INPUT "<ENTER> TO CONTINUE";XX$
 3652 GOTO 340
 4000 IF LEN(Z$(1))=6 THEN RETURN
 4010 FOR J=LEN(Z$(1)) TO 5:Z$(1)=Z$(1)+" ":NEXT J
 4020 RETURN
 4400 LPRINT:LPRINT "===== OVERALL SUMMARY ====="
 4410 LPRINT:LPRINT TAB(11)"NUMBER OF ";
 4415 IF RIGHT$(AM$,1)<>"Y" THEN LPRINT AM$;"S":GOTO 4420
 4416 LPRINT LEFT$(AM$,LEN(AM$)-1);"IES"
 4420 LPRINT "BAND  WORKED  CONFMD  UNCONFMD"
 4430 LPRINT "----  ----    ----     ----"
 4440 GOSUB 4800
 4450 FOR J=1 TO NB
 4470 LPRINT BN$(J);TAB(8)X(1,J);TAB(16)X(2,J);TAB(25)NA-X(2,J)
 4480 NEXT J
 4490 LPRINT "       ----    ----     ----"
 4492 LPRINT "ALL";TAB(8)Y1;TAB(16)Y2;TAB(25)NA-Y2
 4500 LPRINT:LPRINT "TOTAL LOGGED CONTACTS =";LH
 4520 RETURN
 4800 FOR J=1 TO NB:X(1,J)=0:X(2,J)=0:NEXT J
 4810 Y1=0:Y2=0
 4820 FOR RN=1 TO NA
 4830 GOSUB 2440
 4840 Y(1)=0:Y(2)=0
 4850 FOR J=1 TO NB
 4860 IF BS(J)=2 THEN X(2,J)=X(2,J)+1:Y(2)=Y(2)+1
 4870 IF BS(J)>0 THEN X(1,J)=X(1,J)+1:Y(1)=Y(1)+1
 4880 NEXT J
 4890 IF Y(1)>0 THEN Y1=Y1+1
 4900 IF Y(2)>0 THEN Y2=Y2+1
 4910 NEXT RN
 4920 RETURN
```

AWARDS DATA BASE (Apple)

```
10  REM  RADIO AWARDS DATA BASE PROGRAM (APPLE)
15  REM  FOR MULTIBAND DXCC, WAS, WAZ, VUCC AND OTHER AWARDS.
20  REM  BY KC6A AND N6NB
30  REM  (C) 1983 BY HAYDEN BOOK CO.,INC.
40 D$ =  CHR$ (4):BL$ =  CHR$ (7)
50 LP$ = D$ + "PR#1":SCN$ = D$ + "PR#0"
```

```
 60 OP$ = D$ + "OPEN "
 70 RD$ = D$ + "READ "
 80 WR$ = D$ + "WRITE "
 90 CL$ = D$ + "CLOSE "
 100 DL$ = D$ + "DELETE "
 110 DV$ = ",D1"
 120 F1$ = "AWRDINFO"
 130 F2$ = "AWRDBAND"
 140 F3$ = "AWRDCALL":R3$ = ",L37"
 150  HOME : PRINT : INVERSE
 160  PRINT "                                        "
 170  PRINT "      DATA BASE FOR RADIO AWARDS        "
 180  PRINT "                                        "
 190  NORMAL
 200  PRINT OP$;F1$;DV$
 210  PRINT RD$;F1$
 220  INPUT AN$: INPUT AM$
 230  INPUT NB: INPUT NA
 240  DIM BN$(NB),P$(NA),PX$(NA)
 250  DIM BS(NB),RP(NB),X(2,NB)
 260 R2$ = ",L" +  STR$ (7 * NB)
 270  FOR J = 1 TO NB: INPUT BN$(J): NEXT J
 280  FOR J = 1 TO NA
 290  INPUT P$(J),PX$(J)
 300  NEXT J: PRINT CL$
 302  FOR J = 1 TO NB
 303  IF  LEN (BN$(J)) > 4 THEN  PRINT BL$;"TOO MANY CHARACTERS IN ";BN$(J);". MAX
IS 4"
 304  NEXT J
 305  FOR J = 1 TO NA
 306  IF  LEN (P$(J)) > 5 THEN  PRINT BL$;"TOO MANY CHARACTERS IN ";P$(J);". MAX IS
 5"
 307  NEXT J
 310  ONERR  GOTO 3610
 320 RN = 0: GOSUB 2690:LH =  VAL (KL$)
 330  POKE 216,0: REM  RESET ONERR
 340  PRINT : PRINT  TAB( 20 -  LEN (AN$) / 2)AN$
 350  GOSUB 3105: REM  MENU
 360  PRINT : INPUT "ENTER COMMAND? ";C$
 370  IF C$ = "?" THEN  GOSUB 3105: GOTO 360
 380  IF C$ = "SUM" THEN  GOSUB 490: GOTO 360
 390  IF C$ = "STAT" THEN  GOSUB 1580: GOTO 360
 400  IF C$ = "LIST" THEN  GOSUB 1780: GOTO 360
 410  IF C$ = "LOOKUP" THEN  GOSUB 1480: GOTO 360
 420  IF C$ = "QSL" THEN  GOSUB 1110: GOTO 360
 430  IF C$ = "DELETE" THEN  GOSUB 1290: GOTO 360
 440  IF C$ = "QSO" THEN  GOSUB 780: GOTO 360
 450  IF C$ = "QSLSENT" THEN  GOSUB 2300: GOTO 360
 460  IF C$ = "NEW" THEN  GOSUB 3340: GOTO 360
 470  IF C$ = "QRT" THEN  END
 480  PRINT : PRINT BL$;"INVALID COMMAND. PLEASE TRY AGAIN": GOTO 360
 490  PRINT : REM  SUMMARY
 500  PRINT : PRINT "===== OVERALL SUMMARY ====="
 510  PRINT : PRINT  SPC( 6)"NUMBER OF ";
 515  IF  RIGHT$ (AM$,1) <  > "Y" THEN  PRINT AM$;"S": GOTO 520
 516  PRINT  LEFT$ (AM$, LEN (AM$) - 1);"IES"
 520  PRINT "BAND  WORKED  CONFMD  UNCONFMD"
 530  PRINT "----  ----  ----  ----"
 540  FOR J = 1 TO NB:X(1,J) = 0:X(2,J) = 0: NEXT J
 550 Y1 = 0:Y2 = 0
 560  FOR RN = 1 TO NA
 570  GOSUB 2440
 580 Y(1) = 0:Y(2) = 0
 590  FOR J = 1 TO NB
 600  IF BS(J) = 2 THEN X(2,J) = X(2,J) + 1:Y(2) = Y(2) + 1
 610  IF BS(J) > 0 THEN X(1,J) = X(1,J) + 1:Y(1) = Y(1) + 1
 620  NEXT J
 630  IF Y(1) > 0 THEN Y1 = Y1 + 1
 640  IF Y(2) > 0 THEN Y2 = Y2 + 1
 650  NEXT RN
```

```
660  FOR J = 1 TO NB
670  PRINT BN$(J);
680  PRINT  SPC( 7 -  LEN (BN$(J)))X(1,J);
690  PRINT  SPC( 8 -  LEN ( STR$ (X(1,J))))X(2,J);
700  PRINT  SPC( 9 -  LEN ( STR$ (X(2,J))))NA - X(2,J)
710  NEXT J
720  PRINT "          ----      ----      ----"
730  PRINT "ALL       ";Y1;
740  PRINT  SPC( 8 -  LEN ( STR$ (Y1)))Y2;
750  PRINT  SPC( 9 -  LEN ( STR$ (Y2)))NA - Y2
760  PRINT : PRINT "TOTAL LOGGED CONTACTS= ";LH
770  RETURN
780  REM  ADD A CONTACT TO LOG
790  GOSUB 2820: GOSUB 2980
800  IF BS(B) < 2 THEN 840
810  PRINT AM$;" ALREADY CONFIRMED!";BL$
820  INPUT "DO YOU WANT TO ENTER A NEW QSO ANYWAY? (Y/N) ";A$
830  IF A$ <  > "Y" THEN  RETURN
840  INPUT "WHAT IS THE CALL? ";Z$(1): IF  LEN (Z$(1)) > 6 THEN 840
842  IF  LEN (Z$(1)) < 3 THEN 840
850  GOSUB 4000
860  IF BS(B) = 0 THEN 910
870 RN = RP(B)
880  GOSUB 2690
890  IF KL$ = Z$(1) THEN  PRINT "DUPLICATE CONTACT";BL$: GOSUB 3260: GOSUB 3300
900  IF NX > 0 THEN RN = NX: GOTO 880
910  INPUT "WHAT IS THE DATE OF THE QSO? (MM/DD/YY) ";D$
920  GOSUB 2870:Z$(2) = DA$
930  INPUT "WHAT IS THE UTC OF THE QSO? (HHMM) ";Z$(3)
940  IF  LEN (Z$(3)) <  > 4 THEN  PRINT BL$;"UTC MUST BE 4 DIGITS. RE-ENTER.": GOT
O 930
950  INPUT "WHAT WAS THE RST SENT? ";Z$(4)
960  IF  LEN (Z$(4)) > 3 THEN  PRINT BL$;"ERROR. RE-ENTER": GOTO 950
970  INPUT "WHAT WAS THE RST RECEIVED? ";Z$(5)
980  IF  LEN (Z$(5)) > 3 THEN  PRINT BL$;"ERROR. RE-ENTER": GOTO 970
990  INPUT "WHAT WAS THE MODE? ";Z$(6)
1000  IF  LEN (Z$(6)) > 3 THEN Z$(6) =  LEFT$ (Z$(6),3)
1010  INPUT "IS A QSL BEING SENT? (Y/N) ";A$
1020 Z$(7) = "      ": IF A$ = "Y" THEN Z$(7) = Z$(2)
1030 Z$(8) = " ":Z = 0
1040 LH = LH + 1: IF BS(B) > 0 THEN 1080
1050 BS(B) = 1:RP(B) = LH
1060  GOSUB 2490
1070 RN = LH: GOSUB 2660: GOTO 1100
1080 NX = LH: GOSUB 2550
1090 RN = LH: GOSUB 2660
1100 RN = 0: GOSUB 3560: RETURN
1110  REM  CONFIRM A CONTACT
1120  GOSUB 2820: GOSUB 2980:NX = RP(B):S3 = RN
1130  INPUT "WHAT IS THE CALL TO BE CONFIRMED? ";Z$(1)
1140  GOSUB 4000
1150 S4 = 0
1160  IF NX > 0 THEN 1190
1170  IF S4 = 1 THEN  RETURN
1180  PRINT "CALL NOT FOUND";BL$: RETURN
1190 RN = NX: GOSUB 2690
1200  IF Z$(1) <  > KL$ THEN 1160
1210 S4 = 1: GOSUB 3260: GOSUB 3300
1220  PRINT : INPUT "CONFIRM THIS QSO? (Y/N) ";A$
1230  IF A$ <  > "Y" THEN 1160
1240 QR$ = "Y": GOSUB 2550
1250 BS(B) = 2:RN = S3: GOSUB 2490
1260  PRINT : PRINT  TAB( 15)"CONFIRMED"
1270  IF QS$ = "      " THEN  PRINT : FLASH : PRINT BL$;"YOU NEED TO SEND A QSL!":
 NORMAL
1280  RETURN
1290  REM  DELETE A CONTACT
1300  GOSUB 2820: GOSUB 2980:NX = RP(B):S1 = 0:S2 = 0:S3 = RN
1310  INPUT "WHAT IS THE CALL TO BE DELETED? ";Z$(1)
```

```
1320  GOSUB 4000
1330 S4 = 0
1340  IF NX = 0 THEN 1170
1350 RN = NX: GOSUB 2690: IF QR$ = "Y" THEN S2 = S2 + 1
1360  IF  LEFT$ (KL$,1) <  > "*" THEN S1 = S1 + 1
1370  IF Z$(1) <  > KL$ THEN 1340
1380  PRINT : PRINT KL$;" ";DT$;" ";UT$;" ";BN$(B);" ";MD$;" ";QR$
1390  PRINT : INPUT "DELETE THIS QSO? (Y/N) ";A$
1400  IF A$ <  > "Y" THEN S4 = 1: GOTO 1340
1410  IF QR$ = "Y" THEN S2 = S2 - 1
1420 KL$ = "*      ":QR$ = " ": GOSUB 2550
1430  IF NX = 0 THEN 1460
1440 RN = NX: GOSUB 2690: IF QR$ = "Y" THEN S2 = S2 + 1
1450  IF  LEFT$ (KL$,1) <  > "*" THEN S1 = S1 + 1
1460 BS(B) = 0: IF S1 > 1 THEN BS(B) = 1: IF S2 > 0 THEN BS(B) = 2
1470 RN = S3: GOSUB 2490: RETURN
1480  REM  LIST A CALL
1490  GOSUB 2820: GOSUB 2980:NX = RP(B)
1500  PRINT : INPUT "WHAT IS THE CALL? ";Z$(1)
1510  GOSUB 4000
1520 S4 = 0
1530  IF NX = 0 THEN 1170
1540 RN = NX: GOSUB 2690
1550  IF Z$(1) <  > KL$ THEN 1530
1560  IF S4 = 0 THEN  GOSUB 3260
1570  GOSUB 3300:S4 = 1: GOTO 1530
1580  REM   LIST A CATEGORY
1590  GOSUB 2980
1600  PRINT : PRINT "BAND  STATUS"
1610  PRINT "----  ------"
1620  FOR J = 1 TO NB
1630  PRINT BN$(J); TAB( 7)"";: IF BS(J) = 2 THEN  PRINT "CONFIRMED"
1640  IF BS(J) = 1 THEN  PRINT "WORKED"
1650  IF BS(J) = 0 THEN  PRINT "NEED"
1660  NEXT J
1670  PRINT : INPUT "WANT A DISPLAY OF LOG ENTRIES? (Y/N) ";A$: IF A$ <  > "Y" THE
N  RETURN
1680  FOR J = 1 TO NB:NX = RP(J)
1690  IF BS(J) = 0 THEN 1760
1700  PRINT : PRINT "CAT & BAND= ";C$;", ";BN$(J);
1710  IF BS(J) = 1 THEN  PRINT "   NOT CONFIRMED": GOTO 1730
1720  PRINT "   CONFIRMED"
1730  GOSUB 3260
1740  IF NX > 0 THEN RN = NX: GOSUB 2690: GOSUB 3300: GOTO 1740
1750  IF J < NB THEN  PRINT : INPUT "HIT <CR> TO CONTINUE";A$
1760  NEXT J
1770  RETURN
1780  PRINT LP$: REM  LIST ALL CATEGORIES TO PRINTER
1790  PRINT : PRINT "     ***  ";AN$;"  ***"
1800  PRINT : GOSUB 490: PRINT : PRINT
1810  PRINT :K =  INT (NB * 5 / 2) - 1
1820  FOR J = 1 TO K: PRINT "=";: NEXT J: PRINT " SUMMARY BY ";AM$;" ";
1830  FOR J = 1 TO K: PRINT "=";: NEXT J: PRINT : PRINT
1840  PRINT " - = NOT WORKED;  W = WORKED & UNCONFMD;  C = CONFIRMED": PRINT
1850  PRINT AM$;"   BAND"
1860  PRINT "      ";
1870  FOR J = 1 TO NB
1880  PRINT " ";BN$(J); SPC( 4 -  LEN (BN$(J)))"";
1890  NEXT : PRINT " ALL  ";AM$
1900  PRINT "-----  ";: FOR J = 1 TO NB + 1: PRINT "--   ";: NEXT J: PRINT "  ----
------"
1910  FOR JJ = 1 TO NA
1920 RN = JJ: GOSUB 2440: REM  GET F2 RECORD
1930 S1 = 0
1940  PRINT P$(JJ); SPC( 6 -  LEN (P$(JJ)))": ";
1950  FOR J = 1 TO NB
1960  IF BS(J) = 0 THEN  PRINT "-    ";
1970  IF BS(J) = 1 THEN  PRINT "W    ";: IF S1 = 0 THEN S1 = 1
1980  IF BS(J) = 2 THEN  PRINT "C    ";:S1 = 2
1990  NEXT J
```

```
2000  IF S1 = 0 THEN  PRINT "-   ";PX$(JJ)
2010  IF S1 = 1 THEN  PRINT "W   ";PX$(JJ)
2020  IF S1 = 2 THEN  PRINT "C   ";PX$(JJ)
2030  NEXT JJ
2040  PRINT : PRINT : PRINT "============================ LOG ====================
========": PRINT
2050  PRINT AM$; SPC( 9 -  LEN (AM$));
2060  PRINT "BAND  CALL     DATE     UTC SNT RCD MODE QSL-SENT  CFMD"
2070  PRINT "-------- ---- ------ -------- ---- --- --- ---- --------  ----"
2080  PRINT
2090  FOR JJ = 1 TO NA:RN = JJ: GOSUB 2440
2100  FOR KK = 1 TO NB
2110  IF RP(KK) > 0 THEN RN = RP(KK): GOSUB 2140
2120  NEXT KK: NEXT JJ
2130  PRINT : PRINT : PRINT SCN$: RETURN
2140  IF RN = 0 THEN  RETURN
2150  GOSUB 2690: IF  LEFT$ (KL$,1) <  > "*" THEN  GOSUB 2170
2160 RN = NX: GOTO 2140
2170  PRINT P$(JJ);
2180  PRINT  SPC( 9 -  LEN (P$(JJ)))BN$(KK);
2190  PRINT  SPC( 5 -  LEN (BN$(KK)))KL$;
2200  PRINT  SPC( 7 -  LEN (KL$)) LEFT$ (DT$,2);"/"; MID$ (DT$,2,2);"/"; RIGHT$ (D
T$,2);
2210  PRINT  SPC( 1)UT$;
2220  PRINT  SPC( 1)RS$;
2230  PRINT  SPC( 4 -  LEN (RS$))RR$;
2240  PRINT  SPC( 5 -  LEN (RR$))MD$;
2250  PRINT  SPC( 4 -  LEN (MD$))"";
2260  IF QS$ = "      " THEN  PRINT  SPC( 4)QS$;: GOTO 2280
2270  PRINT  LEFT$ (QS$,2);"/"; MID$ (QS$,2,2);"/"; RIGHT$ (QS$,2);"  ";
2280  PRINT "   ";QR$
2290  RETURN
2300  REM  MARK QSL SENT
2310  GOSUB 2820: GOSUB 2980:NX = RP(B)
2320  INPUT "WHAT IS THE CALL TO BE QSL'D? ";Z$(1)
2330  GOSUB 4000
2340 S4 = 0
2350  IF NX = 0 THEN 1170
2360 RN = NX: GOSUB 2690
2370  IF Z$(1) <  > KL$ THEN 2350
2380  GOSUB 3260: GOSUB 3300
2390  PRINT : INPUT "QSL THIS QSO? (Y/N) ";A$
2400  IF A$ <  > "Y" THEN S4 = 1: GOTO 2350
2410  PRINT : INPUT "WHAT IS THE DATE? (MM/DD/YY) ";D$
2420  GOSUB 2870:QS$ = DA$
2430  GOSUB 2550: RETURN
2440  PRINT OP$;F2$;R2$;DV$: REM  READ BAND REC
2450  PRINT RD$;F2$;",R";RN
2460  FOR J = 1 TO NB: INPUT BS(J): INPUT RP(J): NEXT J
2470  PRINT CL$
2480  RETURN
2490  REM  WRITE F2
2500  PRINT OP$;F2$;R2$;DV$
2510  PRINT WR$;F2$;",R"RN
2520  FOR J = 1 TO NB: PRINT BS(J): PRINT RP(J): NEXT J
2530  PRINT CL$
2540  RETURN
2550  REM  WRITE LOG ENTRY
2560  PRINT OP$;F3$;R3$;DV$
2570  PRINT WR$;F3$;",R";RN
2580  PRINT KL$;
2590  PRINT  SPC( 6 -  LEN (KL$))DT$;UT$;RS$;
2600  PRINT  SPC( 3 -  LEN (RS$))RR$;
2610  PRINT  SPC( 3 -  LEN (RR$))MD$;
2620  PRINT  SPC( 3 -  LEN (MD$))QS$;
2630  PRINT  SPC( 6 -  LEN (QS$))QR$;NX;
2640  PRINT  SPC( 4 -  LEN ( STR$ (NX)))""
2650  PRINT CL$: RETURN
2660 KL$ = Z$(1):DT$ = Z$(2):UT$ = Z$(3):RS$ = Z$(4)
2670 RR$ = Z$(5):MD$ = Z$(6):QS$ = Z$(7):QR$ = Z$(8)
```

```
2680 NX = Z: GOTO 2560
2690  PRINT OP$;F3$;R3$;DV$
2700  PRINT RD$;F3$;",R";RN
2710  INPUT A$: PRINT CL$
2720 KL$ =  LEFT$ (A$,6)
2730 DT$ =  MID$ (A$,7,6)
2740 UT$ =  MID$ (A$,13,4)
2750 RS$ =  MID$ (A$,17,3)
2760 RR$ =  MID$ (A$,20,3)
2770 MD$ =  MID$ (A$,23,3)
2780 QS$ =  MID$ (A$,26,6)
2790 QR$ =  MID$ (A$,32,1)
2800 NX =  VAL ( MID$ (A$,33,4))
2810  RETURN
2820  IF NB = 1 THEN B = 1: RETURN
2822  PRINT : INPUT "WHAT BAND? ";B$
2830  FOR B = 1 TO NB: IF B$ = BN$(B) THEN  RETURN
2840  NEXT B
2850  PRINT BL$;"INVALID BAND. VALID BANDS ARE:"
2860  FOR J = 1 TO NB: PRINT  TAB( 5)BN$(J): NEXT J: GOTO 2820
2870  REM  CHECK & COMPRESS DATE
2880  IF  LEN (D$) > 8 THEN 2970
2890  IF  LEN (D$) < 6 THEN 2970
2900  IF  MID$ (D$,2,1) = "/" THEN D$ = "0" + D$
2910  IF  MID$ (D$,3,1) <  > "/" THEN 2970
2920 DA$ =  LEFT$ (D$,2)
2930  IF  MID$ (D$,5,1) = "/" THEN DA$ = DA$ + "0" +  MID$ (D$,4,1): GOTO 2960
2940  IF  MID$ (D$,6,1) <  > "/" THEN 2970
2950 DA$ = DA$ +  MID$ (D$,4,2)
2960 DA$ = DA$ +  RIGHT$ (D$,2): RETURN
2970  INPUT "INVALID DATE. RE-ENTER (MM/DD/YY)? ";D$: GOTO 2880
2980  REM  GET CATEGORY
2990  PRINT : PRINT "WHAT IS THE ";AM$;: INPUT "? ";C$
3000  FOR J = 1 TO NA
3010  IF C$ = P$(J) THEN  PRINT AM$;" IS ";PX$(J): GOTO 3090
3020  NEXT J
3030  PRINT BL$;"INVALID ";AM$;". POSSIBLE VALUES ARE:"
3040  IF  LEN (C$) < 1 THEN 2990
3050 AA$ =  LEFT$ (C$,1)
3060  FOR J = 1 TO NA
3070  IF AA$ =  LEFT$ (P$(J),1) THEN  PRINT  TAB( 5)P$(J); TAB( 15)PX$(J)
3080  NEXT J: GOTO 2990
3090 RN = J: GOSUB 2440: REM  GET BAND RECORD FROM F2
3100  RETURN
3105  HOME : PRINT : INVERSE : PRINT "    RADIO AWARDS DATA BASE PROGRAM     "
3110  PRINT "                                       "
3120  PRINT " '?' CALLS UP THIS MENU                "
3130  PRINT " 'SUM' DISPLAYS AN OVERALL SUMMARY     "
3140  PRINT " 'STAT' SHOWS THE STATUS OF ONE ";AM$;
3150  PRINT  SPC( 7 -  LEN (AM$))""
3160  PRINT " 'LIST' PRINTS FULL SUMMARY AND LOG    "
3170  PRINT " 'LOOKUP' DISPLAYS ONE LOG ENTRY       "
3180  PRINT " 'DELETE' REMOVES A LOG ENTRY          "
3190  PRINT " 'QSO' ADDS A NEW QSO TO THE LOG       "
3200  PRINT " 'QSL' MARKS QSL AS RECEIVED           "
3210  PRINT " 'QSLSENT' MARKS QSL SENT IN THE LOG   "
3220  PRINT " 'NEW' SETS UP FILES FOR A NEW AWARD   "
3230  PRINT " 'QRT' EXITS THE PROGRAM               "
3240  PRINT "                                       "
3250  NORMAL : RETURN
3260  PRINT : REM  DISPLAY A CALL
3270  PRINT " CALL    DATE    UTC SNT RCD MDE QSLSNT C"
3280  PRINT "------ ------ ---- --- --- --- ------ -"
3290  RETURN
3300  IF  LEFT$ (KL$,1) = "*" THEN  RETURN
3310  PRINT KL$; TAB( 8)DT$; TAB( 15)UT$; TAB( 20)RS$;
3320  PRINT  TAB( 24)RR$; TAB( 28)MD$; TAB( 32)QS$; TAB( 39)QR$
3330  RETURN
3340  HOME : PRINT : REM  CREATE F2 & F3 FILES
3350  PRINT BL$; TAB( 15);: FLASH : PRINT " WARNING ": INVERSE
```

```
3360  PRINT "THIS WILL ERASE EXISTING LOG FILES AND "
3370  PRINT "LOGGED CONTACTS TO CREATE A NEW DATA    "
3380  PRINT "BASE.  THIS COMMAND USES A MASTER FILE:"
3390  PRINT "'AWH.INF' (WHICH MUST BE SET UP WITH A "
3400  PRINT "WORD PROCESSOR OR TEXT EDITOR).         "
3410  NORMAL : PRINT : PRINT "DO YOU STILL WANT TO CREATE NEW FILES?": INPUT " Y/N
 ";A$
3420  IF A$ <  > "Y" THEN  RETURN
3422  PRINT : PRINT  TAB( 15)"..WORKING"
3430  PRINT OP$;F2$;R2$;DV$
3440  PRINT DL$;F2$
3450  PRINT OP$;F2$;R2$;DV$
3460  FOR J = 1 TO NA
3470  PRINT WR$;F2$;",R";J
3480  FOR K = 1 TO NB
3490  PRINT 0: PRINT 0
3500  NEXT K: NEXT J
3510  PRINT CL$
3520 LH = 0
3530  GOSUB 3540: PRINT : PRINT  TAB( 17)"-DONE-": GOSUB 3110: RETURN
3540  PRINT OP$;F3$;R3$;DV$
3550  PRINT DL$;F3$
3560  PRINT OP$;F3$;R3$;DV$
3570  PRINT WR$;F3$;",R";0
3580  PRINT LH; SPC( 36 -  LEN ( STR$ (LH)))""
3590  PRINT CL$
3600  RETURN
3610  REM  NO F3 FILE FOUND
3620  PRINT : PRINT BL$; TAB( 15);: FLASH : PRINT " WARNING "
3630  INVERSE : PRINT "DATA FILES ARE MISSING. USE THE 'NEW'  "
3640  PRINT "COMMAND BEFORE PROCEEDING                "
3650  NORMAL : PRINT : GOTO 330
4000  IF  LEN (Z$(1)) = 6 THEN  RETURN
4010  FOR J =  LEN (Z$(1)) TO 5:Z$(1) = Z$(1) + " ": NEXT J
4020  RETURN
```

[redacted] AWARDS DATA BASE (Commodore 64)

```
 10 REM RADIO AWARDS DATA BASE PROGRAM (C64 VERSION)
 20 REM FOR MULTIBAND DXCC, WAS, WAZ, VUCC AND OTHER AWARDS.
 30 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO., INC.
 32 PRINT:PRINT:OPEN15,8,15
 35 OPEN130,2,5,CHR$(136)+CHR$(1):REM RS232 PRINTER CHANNEL
 40 C7$=CHR$(147):PRINT C7$:BL$=CHR$(7):HI$=CHR$(18):LO$=CHR$(146)
 60 REM DATA FILE (AWRDINFO) MUST BE ON LOGGED DISK DRIVE
100 F1$="0:AWRDINFO":F2$="0:AWRDBAND":F3$="0:AWRDCALL":R3=34
150 PRINT:PRINT:PRINT " ";HI$;"                                    "
160 PRINT " ";HI$;"     DATA BASE FOR RADIO AWARDS        "
165 PRINT " ";HI$;"                                       "
170 PRINT LO$:PRINT:PRINT
200 OPEN4,8,4,F1$+",S,R":GOSUB 6000
205 INPUT#15,EA,EB$,EC,ED:IF EA<20 THEN 220
210 IF EA=62 THEN 3610
220 INPUT#4,AN$:INPUT#4,AM$:INPUT#4,NB:INPUT#4,NA
240 DIM BN$(NB),P$(NA),PX$(NA),BS(NB),RP(NB),X(2,NB),BX(NA,NB),RX(NA,NB):R2=4
270 FOR J=1 TO NB:INPUT#4,BN$(J)
271 IF LEN(BN$(J))>4 THEN PRINT BL$;"TOO MANY CHARS IN ";BN$(J);". MAX IS 4"
272 NEXT J
280 FOR J=1 TO NA
290 INPUT#4,P$(J),PX$(J)
292 IF LEN(P$(J))>5 THEN PRINT BL$;"TOO MANY CHARS IN ";P$(J);". MAX IS 5"
300 NEXT J:CLOSE4
302 PRINT "DO YOU WANT TO USE EXISTING LOGS (E)"
```

```
304 PRINT "OR CREATE NEW ONES?"
306 INPUT "(E) OR (N)";A1$
307 IF A1$="N" THEN GOSUB 3340:GOTO 360
308 IF A1$<>"E"THEN PRINT:PRINT "-INVALID ENTRY-":PRINT:GOTO 302
310 OPEN4,8,4,F2$+",S,R":GOSUB 6000
311 FOR J=1 TO NA:FOR K=1 TO NB
312 INPUT#4,BX(J,K):INPUT#4,RX(J,K):NEXT K:NEXT J
315 CLOSE4:GOSUB 6000
320 RN=0:GOSUB 2690:LH=VAL(KL$)
330 PRINT:PRINT TAB((40-LEN(AN$))/2)AN$
350 GOSUB 3110:REM MENU
360 PRINT:PRINT HI$;"ENTER COMMAND";:INPUT C$:PRINT LO$
370 IF C$="?" THEN PRINT C7$:GOSUB 3110:GOTO 360
380 IF C$="SUM" THEN GOSUB 490:GOTO 360
390 IF C$="STAT" THEN GOSUB 1580:GOTO 360
400 IF C$="LIST" THEN GOSUB 1780:GOTO 360
410 IF C$="LOOKUP" THEN GOSUB 1480:GOTO  360
420 IF C$="QSL" THEN GOSUB 1110:GOTO 360
430 IF C$="DELETE" THEN GOSUB 1290:GOTO 360
440 IF C$="QSO" THEN GOSUB 780:GOTO 360
450 IF C$="QSLSENT" THEN GOSUB 2300:GOTO 360
470 IF C$="QRT" THEN GOTO 5000
480 PRINT:PRINT BL$;"INVALID COMMAND. PLEASE TRY AGAIN":GOTO 360
490 PRINT:REM SUMMARY
500 PRINT:PRINT "===== OVERALL SUMMARY ====="
510 PRINT:PRINT TAB(6)"NUMBER OF ";
515 IF RIGHT$(AM$,1)<>"Y" THEN PRINT AM$;"S":GOTO 520
516 PRINT LEFT$(AM$,LEN(AM$)-1);"IES"
520 PRINT "BAND  WORKED  CONFMD  UNCONFMD"
530 PRINT "----  ----    ----    ----"
540 GOSUB 4800
660 FOR J=1 TO NB
670 PRINT BN$(J);TAB(8)X(1,J);TAB(16)X(2,J);TAB(25)NA-X(2,J)
710 NEXT J
720 PRINT "      ----    ----    ----"
730 PRINT "ALL    ";Y1;TAB(16)Y2;TAB(25)NA-Y2
760 PRINT:PRINT "TOTAL LOGGED CONTACTS=";LH
770 RETURN
780 REM ADD A CONTACT TO LOG
790 GOSUB 2820:GOSUB 2980
800 IF BS(B)<2 THEN 840
810 PRINT AM$;" ALREADY CONFIRMED!";BL$
820 PRINT "DO YOU WANT TO ENTER A NEW QSO ANYWAY (Y/N)":INPUT A$
830 IF A$<>"Y" THEN RETURN
840 Z$(1)="":INPUT "WHAT IS THE CALL";Z$(1):IF LEN(Z$(1))>6 THEN 840
842 IF LEN(Z$(1))<3 THEN 840
850 GOSUB 4000
860 IF BS(B)=0 THEN 910
870 RN=RP(B)
880 GOSUB 2690
890 IF KL$=Z$(1) THEN PRINT "DUPLICATE CONTACT";BL$:GOSUB 3260:GOSUB 3300
900 IF NX>0 THEN RN=NX:GOTO 880
910 D$="":PRINT "WHAT IS THE DATE OF THE QSO (MM/DD/YY)":INPUT D$
920 GOSUB 2870:Z$(2)=DA$
930 PRINT "WHAT IS THE UTC OF THE QSO (HHMM)":INPUT Z$(3)
940 IF LEN(Z$(3))<>4 THEN PRINT BL$;"UTC MUST BE 4 DIGITS. RE-ENTER.":GOTO 930
950 Z$(4)="-":PRINT "WHAT IS THE RST SENT";:INPUT Z$(4)
960 IF LEN(Z$(4))>3 THEN PRINT BL$;"ERROR. RE-ENTER":GOTO 950
970 Z$(5)="-":PRINT "WHAT WAS THE RST RECEIVED";:INPUT Z$(5)
980 IF LEN(Z$(5))>3 THEN PRINT BL$;"ERROR. RE-ENTER":GOTO 970
990 Z$(6)="-":PRINT "WHAT WAS THE MODE";:INPUT Z$(6)
1000 IF LEN(Z$(6))>3 THEN Z$(6)=LEFT$(Z$(6),3)
1010 INPUT "IS A QSL BEING SENT (Y/N)";A$
1020 Z$(7)="N     ":IF A$="Y"  THEN Z$(7)=Z$(2)
1030 Z$(8)="N":Z=0
1040 LH=LH+1:IF BS(B)>0 THEN 1080
1050 BS(B)=1:RP(B)=LH
1060 GOSUB 2490
1070 RN=LH:GOSUB 2660:GOTO 1100
1080 NX=LH:GOSUB 2550
```

```
1090 RN=LH:GOSUB 2660
1100 RN=0:GOSUB 3540:RETURN
1110 REM CONFIRM A CONTACT
1120 GOSUB 2820:GOSUB 2980:NX=RP(B):S3=RN
1130 PRINT "WHAT IS THE CALL TO BE CONFIRMED":INPUT Z$(1)
1140 GOSUB 4000:S4=0
1160 IF NX>0 THEN 1190
1170 IF S4=1 THEN RETURN
1180 PRINT "CALL NOT FOUND";BL$:RETURN
1190 RN=NX:GOSUB 2690
1200 IF Z$(1)<>KL$ THEN 1160
1210 S4=1:GOSUB 3260:GOSUB 3300
1220 PRINT:INPUT "CONFIRM THIS QSO (Y/N) ";A$
1230 IF A$<>"Y" THEN 1160
1240 QR$="Y":GOSUB 2550
1250 BS(B)=2:RN=S3:GOSUB 2490
1260 PRINT:PRINT TAB(15)"CONFIRMED"
1270 IF LEFT$(QS$,1)="N" THEN PRINT:PRINT BL$;"YOU NEED TO SEND A QSL!"
1280 RETURN
1290 REM DELETE A CONTACT
1300 GOSUB 2820:GOSUB 2980:NX=RP(B):S1=0:S2=0:S3=RN
1310 PRINT "WHAT IS THE CALL TO BE DELETED":INPUT Z$(1)
1320 GOSUB 4000:S4=0
1340 IF NX=0 THEN 1170
1350 RN=NX:GOSUB 2690:IF QR$="Y" THEN S2=S2+1
1360 IF LEFT$(KL$,1)<>"*" THEN S1=S1+1
1370 IF Z$(1)<>KL$ THEN 1340
1380 PRINT:PRINT KL$;" ";DT$;" ";UT$;" ";BN$(B);" ";MD$;" ";QR$
1390 PRINT:INPUT "DELETE THIS QSO (Y/N) ";A$
1400 IF A$<>"Y" THEN S4=1:GOTO 1340
1410 IF QR$="Y" THEN S2=S2-1
1420 KL$="*     ":QR$=" ":GOSUB 2550
1430 IF NX=0 THEN 1460
1440 RN=NX:GOSUB 2690:IF QR$="Y" THEN S2=S2+1
1450 IF LEFT$(KL$,1)<>"*" THEN S1=S1+1
1460 BS(B)=0:IF S1>1 THEN BS(B)=1:IF S2>0 THEN BS(B)=2
1470 RN=S3:GOSUB 2490:RETURN
1480 REM LIST A CALL
1490 GOSUB 2820:GOSUB 2980:NX=RP(B)
1500 PRINT:INPUT "WHAT IS THE CALL ";Z$(1)
1510 GOSUB 4000:S4=0
1530 IF NX=0 THEN 1170
1540 RN=NX:GOSUB 2690
1550 IF Z$(1)<>KL$ THEN 1530
1560 IF S4=0 THEN GOSUB 3260
1570 GOSUB 3300:S4=1:GOTO 1530
1580 REM LIST A CATEGORY
1590 GOSUB 2980:PRINT:PRINT "BAND  STATUS":PRINT "----  ------"
1620 FOR J=1 TO NB
1630 PRINT BN$(J);TAB(7)"";:IF BS(J)=2 THEN PRINT "CONFIRMED"
1640 IF BS(J)=1 THEN PRINT "WORKED"
1650 IF BS(J)=0 THEN PRINT "NEED"
1660 NEXT J
1670 PRINT:PRINT "WANT A DISPLAY OF LOG ENTRIES (Y/N) ":INPUT A$
1671 IF A$<>"Y" THEN RETURN
1680 FOR J=1 TO NB:NX=RP(J)
1690 IF BS(J)=0 THEN 1760
1700 PRINT:PRINT "CAT & BAND= ";C$;", ";BN$(J);
1710 IF BS(J)=1 THEN PRINT "   NOT CONFIRMED":GOTO 1730
1720 PRINT "   CONFIRMED"
1730 GOSUB 3260
1740 IF NX>0 THEN RN=NX:GOSUB 2690:GOSUB 3300:GOTO 1740
1750 IF J<NB THEN PRINT:INPUT "HIT <CR> TO CONTINUE";A$
1760 NEXT J
1770 RETURN
1780 REM LIST ALL CATEGORIES TO PRINTER
1790 PRINT#130,"":PRINT#130,"     ***  ";AN$;"  ***"
1800 PRINT#130,"":GOSUB 4400:PRINT#130,"":PRINT#130,""
1810 PRINT#130,"":K=INT(NB*5/2)-1
1820 FOR J=1 TO K:PRINT#130,"=";:NEXT J:PRINT#130," SUMMARY BY ";AM$;" ";
```

```
1830 FOR J=1 TO K:PRINT#130,"=";:NEXT J:PRINT#130,"":PRINT#130,""
1840 PRINT#130," - = NOT WORKED;  W = WORKED & UNCONFMD;  C = CONFIRMED"
1850 PRINT#130,"":PRINT#130,AM$;"              BAND"
1860 PRINT#130,"          ";
1870 FOR J=1 TO NB
1880 PRINT#130," ";BN$(J);SPC(4-LEN(BN$(J)))"";
1890 NEXT:PRINT#130," ALL  ";AM$
1900 PRINT#130,"--------  ";:FOR J=1 TO NB+1:PRINT#130,"--   ";:NEXT J
1901 PRINT#130,"----------"
1910 FOR JJ=1 TO NA
1920 RN=JJ:GOSUB 2440:REM GET F2 REC
1940 S1=0:PRINT#130, P$(JJ);TAB(7)": ";
1950 FOR J=1 TO NB
1960 IF BS(J)=0 THEN PRINT#130,"-    ";
1970 IF BS(J)=1 THEN PRINT#130,"W    ";:IF S1=0 THEN S1=1
1980 IF BS(J)=2 THEN PRINT#130,"C    ";:S1=2
1990 NEXT J
2000 IF S1=0 THEN PRINT#130,"-   ";PX$(JJ)
2010 IF S1=1 THEN PRINT#130,"W   ";PX$(JJ)
2020 IF S1=2 THEN PRINT#130,"C   ";PX$(JJ)
2030 NEXT JJ
2040 PRINT#130,"":PRINT#130,""
2041 PRINT#130,"============================ LOG ============================"
2042 PRINT#130,""
2050 PRINT#130,AM$;"   BAND  CALL    DATE    UTC SNT RCD MODE QSL-SENT  CFMD"
2070 PRINT#130,"-------- ---- ------ -------- ---- --- --- ---- -------- ----"
2080 PRINT#130,""
2090 FOR JJ=1 TO NA:RN=JJ:GOSUB 2440
2100 FOR KK=1 TO NB
2110 IF RP(KK)>0 THEN RN=RP(KK):GOSUB 2140
2120 NEXT KK:NEXT JJ
2130 PRINT#130,"":PRINT#130,"":RETURN
2140 IF RN=0 THEN RETURN
2150 GOSUB 2690:IF LEFT$(KL$,1)<>"*" THEN GOSUB 2170
2160 RN=NX:GOTO 2140
2170 PRINT#130,"";P$(JJ);
2180 PRINT#130,"";SPC(9-LEN(P$(JJ)))BN$(KK);
2190 PRINT#130,"";SPC(5-LEN(BN$(KK)))KL$;
2200 PRINT#130,"";SPC(7-LEN(KL$))LEFT$(DT$,2);"/";MID$(DT$,2,2);"/";
2202 PRINT#130,"";RIGHT$(DT$,2);
2210 PRINT#130,"";SPC(1)UT$;SPC(1)RS$;
2230 PRINT#130,"";SPC(4-LEN(RS$))RR$;SPC(5-LEN(RR$))MD$;SPC(4-LEN(MD$))"";
2260 IF LEFT$(QS$,1)="N" THEN PRINT#130,"    ";QS$;:GOTO 2280
2270 PRINT#130,"";LEFT$(QS$,2);"/";MID$(QS$,2,2);"/";RIGHT$(QS$,2);"  ";
2280 PRINT#130,"  ";QR$:RETURN
2300 REM MARK QSL SENT
2310 GOSUB 2820:GOSUB 2980:NX=RP(B)
2320 PRINT "WHAT IS THE CALL TO BE QSL'D ":INPUT Z$(1)
2330 GOSUB 4000:S4=0
2350 IF NX=0 THEN 1170
2360 RN=NX:GOSUB 2690
2370 IF Z$(1)<>KL$ THEN 2350
2380 GOSUB 3260:GOSUB 3300
2390 PRINT:INPUT "QSL THIS QSO (Y/N) ";A$
2400 IF A$<>"Y" THEN S4=1:GOTO 2350
2410 PRINT:PRINT "WHAT IS THE DATE (MM/DD/YY) ":INPUT D$
2420 GOSUB 2870:QS$=DA$
2430 GOSUB 2550:RETURN
2440 FOR JK=1 TO NB:BS(JK)=BX(RN,JK):RP(JK)=RX(RN,JK):NEXT JK
2480 RETURN
2490 REM WRITE FILE 2
2500 FOR JK=1 TO NB:BX(RN,JK)=BS(JK):RX(RN,JK)=RP(JK):NEXT JK
2540 CLOSE2:RETURN
2550 REM WRITE LOG ENTRY
2560 OPEN3,8,3,F3$:GOSUB 6000:R1=RN+1:GOSUB 7000
2570 R9=1:GOSUB 7100:PRINT#3,KL$
2580 R9=8:GOSUB 7100:PRINT#3,DT$
2590 R9=15:GOSUB 7100:PRINT#3,UT$
2595 R9=20:GOSUB 7100:PRINT#3,RS$
2600 R9=24:GOSUB 7100:PRINT#3,RR$
```

```
2610 R9=28:GOSUB 7100:PRINT#3,MD$
2620 R9=32:GOSUB 7100:PRINT#3,QS$
2630 R9=39:GOSUB 7100:PRINT#3,QR$
2635 R9=41:GOSUB 7100:PRINT#3,NX
2640 GOSUB 6000
2650 CLOSE3:RETURN
2660 KL$=Z$(1):DT$=Z$(2):UT$=Z$(3):RS$=Z$(4)
2670 RR$=Z$(5):MD$=Z$(6):QS$=Z$(7):QR$=Z$(8)
2680 NX=Z:GOTO 2550
2690 OPEN3,8,3,F3$:GOSUB 6000:R1=RN+1:GOSUB 7000
2691 R9=1:GOSUB 7100:INPUT#3,KL$
2692 R9=8:GOSUB 7100:INPUT#3,DT$
2693 R9=15:GOSUB 7100:INPUT#3,UT$
2694 R9=20:GOSUB 7100:INPUT#3,RS$
2695 R9=24:GOSUB 7100:INPUT#3,RR$
2696 R9=28:GOSUB 7100:INPUT#3,MD$
2697 R9=32:GOSUB 7100:INPUT#3,QS$
2698 R9=39:GOSUB 7100:INPUT#3,QR$
2699 R9=41:GOSUB 7100:INPUT#3,NX
2700 GOSUB 6000
2710 CLOSE3
2810 RETURN
2820 IF NB=1 THEN B=1:RETURN
2822 PRINT:INPUT "WHAT BAND ";B$
2830 FOR B=1 TO NB:IF B$=BN$(B) THEN RETURN
2840 NEXT B
2850 PRINT BL$;"INVALID BAND. VALID BANDS ARE:"
2860 FOR J=1 TO NB:PRINT TAB(5)BN$(J):NEXT J:GOTO 2820
2870 REM CHECK & COMPRESS DATE
2880 IF LEN(D$)>8 THEN 2970
2890 IF LEN(D$)<6 THEN 2970
2900 IF MID$(D$,2,1)="/" THEN D$="0"+D$
2910 IF MID$(D$,3,1)<>"/" THEN 2970
2920 DA$=LEFT$(D$,2)
2930 IF MID$(D$,5,1)="/" THEN DA$=DA$+"0"+MID$(D$,4,1):GOTO 2960
2940 IF MID$(D$,6,1)<>"/" THEN 2970
2950 DA$=DA$+MID$(D$,4,2)
2960 DA$=DA$+RIGHT$(D$,2):RETURN
2970 PRINT "INVALID DATE. RE-ENTER (MM/DD/YY) ":INPUT D$:GOTO 2880
2980 REM GET CATEGORY
2990 PRINT:PRINT "WHAT IS THE ";AM$;:INPUT " ";C$
3000 FOR J=1 TO NA
3010 IF C$=P$(J) THEN PRINT AM$;" IS ";PX$(J):GOTO 3090
3020 NEXT J
3030 PRINT BL$;"INVALID ";AM$;". POSSIBLE VALUES ARE:"
3040 IF LEN(C$)<1 THEN 2990
3050 AA$=LEFT$(C$,1)
3060 FOR J=1 TO NA
3070 IF AA$=LEFT$(P$(J),1) THEN PRINT TAB(5)P$(J);TAB(15)PX$(J)
3080 NEXT J:GOTO 2990
3090 RN=J:GOSUB 2440:REM GET BAND REC FROM F2
3100 RETURN
3110 PRINT:PRINT
3111 PRINT " ";HI$;"    RADIO AWARDS DATA BASE PROGRAM     "
3112 PRINT " ";HI$;"        HERE ARE YOUR OPTIONS:         ":PRINT LO$
3120 PRINT " '?' CALLS UP THIS MENU"
3130 PRINT " 'SUM' DISPLAYS AN OVERALL SUMMARY"
3140 PRINT " 'STAT' SHOWS STATUS OF ONE ";AM$
3160 PRINT " 'LIST' PRINTS FULL SUMMARY AND LOG"
3170 PRINT " 'LOOKUP' DISPLAYS ONE LOG ENTRY"
3180 PRINT " 'DELETE' REMOVES A LOG ENTRY"
3190 PRINT " 'QSO' ADDS A NEW QSO TO THE LOG"
3200 PRINT " 'QSL' MARKS QSL AS RECEIVED"
3210 PRINT " 'QSLSENT' MARKS QSL SENT IN THE LOG"
3230 PRINT " 'QRT' EXITS THE PROGRAM"
3240 PRINT
3250 RETURN
3260 PRINT:REM DISPLAY A CALL
3270 PRINT " CALL   DATE   UTC SNT RCD MDE QSLSNT C"
3280 PRINT "------ ------ ---- --- --- --- ------ -"
```

```
3290 RETURN
3300 IF LEFT$(KL$,1)="*" THEN RETURN
3310 PRINT KL$;TAB(8)DT$;TAB(15)UT$;TAB(20)RS$;
3320 PRINT TAB(24)RR$;TAB(28)MD$;TAB(32)QS$;TAB(39)QR$
3330 RETURN
3340 PRINT C7$:PRINT:REM CREATE F2 & F3 FILES
3350 PRINT BL$;TAB(15)"WARNING"
3360 PRINT "THIS WILL ERASE EXISTING LOG FILES AND"
3370 PRINT "LOGGED CONTACTS TO CREATE A NEW DATA"
3380 PRINT "BASE. THIS COMMAND USES A MASTER FILE:"
3390 PRINT "'AWRDINFO' (WHICH MUST BE SET UP FIRST"
3400 PRINT "USING THE 'AWRDINFOSETUP' PROGRAM)"
3410 PRINT:PRINT "DO YOU STILL WANT TO CREATE NEW FILES?"
3412 INPUT "ENTER Y OR N";A$
3420 IF A$<>"Y" THEN RETURN
3422 PRINT:PRINT TAB(15)"...WORKING"
3423 PRINT#15,"S"+F2$:PRINT#15,"S"+F3$
3425 OPEN3,8,3,F3$+",L,"+CHR$(47):CLOSE3
3430 FOR J=1 TO NB:BS(J)=0:RP(J)=0:NEXT J
3460 FOR RN=1 TO NA:GOSUB 2500:NEXT RN
3485 CLOSE2:PRINT:PRINT "FILE #2 INITIALIZED"
3520 LH=0:RN=0:NX=0
3530 GOSUB 3540:PRINT:PRINT TAB(17)"-DONE-":GOSUB 3110:RETURN
3540 FOR J=2 TO 8:Z$(J)="9":NEXT J
3550 Z$(1)=STR$(LH):GOSUB 2660
3600 RETURN
3610 REM ONE OR MORE DATA FILES MISSING
3620 PRINT:PRINT SPC(10)HI$;"     WARNING     "
3630 PRINT:PRINT "AT LEAST ONE OF THE THREE DATA FILES"
3632 PRINT "IS MISSING.  BE SURE THE 'AWRDINFO'"
3634 PRINT "FILE HAS BEEN CREATED.  IF IT HAS, THEN"
3636 PRINT "USE THE 'NEW' COMMAND TO SET UP OTHERS"
3640 PRINT "BEFORE PROCEEDING FURTHER."
3645 PRINT:X3$="":INPUT "ENTER <CR> TO CONTINUE";X3$
3650 PRINT:GOTO 330
4000 IF LEN(Z$(1))=6 THEN RETURN
4010 FOR J=LEN(Z$(1)) TO 5:Z$(1)=Z$(1)+" ":NEXT J
4020 RETURN
4400 PRINT#130,"":PRINT#130,"===== OVERALL SUMMARY ====="
4410 PRINT#130,"":PRINT#130," ";SPC(10)"NUMBER OF ";
4415 IF RIGHT$(AM$,1)<>"Y" THEN PRINT#130,AM$;"S":GOTO 4420
4416 PRINT#130,"";LEFT$(AM$,LEN(AM$)-1);"IES"
4420 PRINT#130,"BAND  WORKED  CONFMD  UNCONFMD"
4430 PRINT#130,"----   ----    ----     ----"
4440 GOSUB 4800
4450 FOR J=1 TO NB
4470 PRINT#130,BN$(J);SPC(8-LEN(BN$(J)))X(1,J);SPC(8-LEN(STR$(X(1,J))))X(2,J);
4472 PRINT#130,"";SPC(9-LEN(STR$(X(2,J))))NA-X(2,J)
4480 NEXT J
4490 PRINT#130,"       ----    ----     ----"
4492 PRINT#130,"ALL     ";Y1;SPC(8-LEN(STR$(Y1)))Y2;SPC(9-LEN(STR$(Y2)))NA-Y2
4500 PRINT#130,"":PRINT#130,"TOTAL LOGGED CONTACTS =";LH
4520 RETURN
4800 FOR J=1 TO NB:X(1,J)=0:X(2,J)=0:NEXT J
4810 Y1=0:Y2=0
4820 FOR RN=1 TO NA
4830 GOSUB 2440
4840 Y(1)=0:Y(2)=0
4850 FOR J=1 TO NB
4860 IF BS(J)=2 THEN X(2,J)=X(2,J)+1:Y(2)=Y(2)+1
4870 IF BS(J)>0 THEN X(1,J)=X(1,J)+1:Y(1)=Y(1)+1
4880 NEXT J
4890 IF Y(1)>0 THEN Y1=Y1+1
4900 IF Y(2)>0 THEN Y2=Y2+1
4910 NEXT RN
4920 RETURN
5000 PRINT#15,"S"+F2$
5005 OPEN4,8,4,F2$+",S,W"
5010 FOR J=1 TO NA:FOR K=1 TO NB:PRINT#4,BX(J,K):PRINT#4,RX(J,K)
5030 NEXT K:NEXT J
```

```
5040 CLOSE4:CLOSE130:CLOSE15:END
6000 INPUT#15,EA,EB$,EC,ED:IF EA<20 THEN RETURN:REM READ ERROR CHANNEL
6010 IF EA=50 THEN RETURN
6020 IF EA=63 THEN RETURN
6030 PRINT EA;EB$;EC;ED:STOP:RETURN
7000 RH=INT(R1/256):RL=R1-256*RH:RETURN
7100 PRINT#15,"P"CHR$(3)CHR$(RL)CHR$(RH)CHR$(R9)
7110 GOTO 6000
7200 RH=INT(RO/256):RL=RO-256*RH:RETURN
7300 PRINT#15,"P"CHR$(2)CHR$(RL)CHR$(RH)CHR$(R9)
7310 GOTO 6000
```

Setup program for AWARDS DATA BASE (TRS-80 and CP/M)

```
10 REM SETUP PROGRAM FOR AWARD INFO DATA FILE (TRS-80)
20 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO., INC.
30 CLS:PRINT "WARNING: THIS PROGRAM DELETES THE"
40 PRINT "'AWRDINFO' FILE FOR THE AWARDS"
50 PRINT "DATE BASE PROGRAM AND CREATES A NEW ONE"
60 PRINT:A$="":INPUT "OK TO CONTINUE (Y/N)";A$
70 IF A$<>"Y" THEN END
75 CLEAR 10000
80 OPEN "O",1,"AWRDINFO"
110 PRINT:PRINT "ENTER NAME OF THIS OPERATING AWARD"
112 INPUT "EXAMPLE- FIVE BAND DXCC";AN$
114 PRINT#1,AN$
120 PRINT:PRINT "ENTER THE NAME OF THE AWARD UNIT"
122 INPUT "EXAMPLE- COUNTRY";AM$
124 PRINT#1,AM$
130 PRINT:PRINT "ENTER THE NUMBER OF SEPARATE BANDS"
132 INPUT "EXAMPLE- 5";NB
134 PRINT#1,NB
140 PRINT:PRINT "ENTER THE NUMBER OF DIFFERENT AWARD"
142 PRINT "UNITS (E.G. STATES OR COUNTRIES)"
144 INPUT NA
146 PRINT#1,NA
150 PRINT:PRINT "NOW, ENTER THE NAMES OF THE BANDS"
152 DIM BN$(NB),P$(NA),NM$(NA)
160 FOR J=1 TO NB
170 PRINT "ENTER THE NAME OF BAND #";J
172 INPUT BN$(J)
174 IF LEN(BN$(J))>4 THEN PRINT "TOO LONG.  4 MAX":GOTO 170
176 PRINT#1,BN$(J)
180 NEXT J
190 PRINT:PRINT "FINALLY, ENTER THE STANDARD PREFIX"
192 PRINT "OR ABBREVIATION FOR EACH ";AM$
194 PRINT "AND ITS NAME (E.G. JA,JAPAN)"
200 FOR J=1 TO NA
210 PRINT "ENTER THE PREFIX AND NAME OF"
212 PRINT AM$;" #";J
214 INPUT P$(J),NM$(J)
220 A$=P$(J)+","+NM$(J)
224 PRINT#1,A$
230 NEXT J
240 CLOSE:END
```

Setup program for AWARDS DATA BASE (Apple)

```
10  REM  SETUP PROGRAM FOR AWARD INFO DATA FILE (APPLE)
20  REM  BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO., INC.
25  HOME
30  PRINT "WARNING: THIS PROGRAM DELETES THE"
40  PRINT "'AWRDINFO' FILE FOR THE AWARDS"
50  PRINT "DATE BASE PROGRAM AND CREATES A NEW ONE"
60  PRINT :A$ = "": INPUT "OK TO CONTINUE (Y/N)";A$
70  IF A$ <  > "Y" THEN  END
80 D$ =  CHR$ (4)
110  PRINT : PRINT "ENTER NAME OF THIS OPERATING AWARD"
112  INPUT "EXAMPLE - FIVE BAND DXCC ";AN$
120  PRINT : PRINT "ENTER THE NAME OF THE AWARD UNIT"
122  INPUT "EXAMPLE- COUNTRY ";AM$
130  PRINT : PRINT "ENTER THE NUMBER OF SEPARATE BANDS"
132  INPUT "EXAMPLE- 5 ";NB
140  PRINT : PRINT "ENTER THE NUMBER OF DIFFERENT AWARD"
142  PRINT "UNITS (E.G. STATES OR COUNTRIES)"
144  INPUT NA
150  PRINT : PRINT "NOW, ENTER THE NAMES OF THE BANDS"
152  DIM BN$(NB),P$(NA),NM$(NA)
160  FOR J = 1 TO NB
170  PRINT "ENTER THE NAME OF BAND #";J
172  INPUT BN$(J)
174  IF  LEN (BN$(J)) > 4 THEN  PRINT "TOO LONG.  4 MAX": GOTO 170
180  NEXT J
190  PRINT : PRINT "FINALLY, ENTER THE STANDARD PREFIX"
192  PRINT "OR ABBREVIATION FOR EACH ";AM$
194  PRINT "AND ITS NAME (E.G. JA,JAPAN)"
200  FOR J = 1 TO NA
210  PRINT "ENTER THE PREFIX AND NAME OF"
212  PRINT AM$;" #";J
214  INPUT P$(J),NM$(J)
230  NEXT J
240  PRINT D$;"OPEN AWRDINFO": PRINT D$;"WRITE AWRDINFO"
250  REM
254  PRINT AN$
264  PRINT AM$
274  PRINT NB
284  PRINT NA
290  FOR J = 1 TO NB
294  PRINT BN$(J)
300  NEXT J
302  FOR J = 1 TO NA
304 A$ = P$(J) + "," + NM$(J)
306  PRINT A$
310  NEXT J
320  PRINT D$;"CLOSE AWRDINFO"
330  END
```

Setup program for AWARDS DATA BASE (Commodore 64)

```
10 REM SETUP PROGRAM FOR AWARD INFO DATA FILE (C64)
20 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO., INC.
30 PRINT "WARNING: THIS PROGRAM DELETES THE"
40 PRINT "'AWRDINFO' FILE FOR THE AWARDS"
50 PRINT "DATE BASE PROGRAM AND CREATES A NEW ONE"
60 PRINT:INPUT "OK TO CONTINUE (Y/N)";A$
70 IF A$<>"Y" THEN END
75 CM$=","
80 OPEN15,8,15
90 PRINT#15,"S0:AWRDINFO"
```

```
100 OPEN4,8,4,"AWRDINFO,S,W"
110 PRINT:PRINT "ENTER NAME OF THIS OPERATING AWARD"
112 INPUT "EXAMPLE- FIVE BAND DXCC";AN$
114 PRINT#4,AN$
120 PRINT:PRINT "ENTER THE NAME OF THE AWARD UNIT"
122 INPUT "EXAMPLE- COUNTRY";AM$
124 PRINT#4,AM$
130 PRINT:PRINT "ENTER THE NUMBER OF SEPARATE BANDS"
132 INPUT "EXAMPLE- 5";NB
134 PRINT#4,NB
140 PRINT:PRINT "ENTER THE NUMBER OF DIFFERENT AWARD"
142 PRINT "UNITS (E.G. STATES OR COUNTRIES)"
144 INPUT NA
146 PRINT#4,NA
150 PRINT "NOW, ENTER THE NAMES OF THE BANDS"
152 DIM BN$(NB),P$(NA),NM$(NA)
160 FOR J=1 TO NB
170 PRINT "ENTER THE NAME OF BAND #";J
172 INPUT BN$(J)
174 IF LEN(BN$(J))>4 THEN PRINT "TOO LONG.  4 MAX":GOTO 170
176 PRINT#4,BN$(J)
180 NEXT J
190 PRINT "FINALLY, ENTER THE STANDARD PREFIX"
192 PRINT "OR ABBREVIATION FOR EACH ";AM$
194 PRINT "AND ITS NAME (E.G. JA,JAPAN)"
200 FOR J=1 TO NA
210 PRINT "ENTER THE PREFIX AND NAME OF"
212 PRINT AM$;" #";J
214 INPUT P$(J),NM$(J)
224 PRINT#4,P$(J)CM$NM$(J)
230 NEXT J
240 CLOSE4:CLOSE15
```

THE GRIDLOCATOR PROGRAM

What it does

The GRIDLOCATOR is a short program that converts the latitude and longitude of any location (except the North and South Poles) into the standard QTH Grid Locators used in the VHF-UHF Century Club awards program and in some VHF-UHF contests.

Grid Locators are used by VHFers in many parts of the world. In Europe, the four-character locator is expanded to six characters, allowing a station's geographic location to be pinpointed within 2.5 minutes latitude and 5 minutes longitude. Because only the first four characters are used for awards and contests in North America at this writing, the additional characters are not included in this program.

Setup instructions

This program will run as written on most kinds of microcomputers, with or without disk drives or a printer. It uses no function or command that is not generally accepted as good syntax on most systems, including TRS-80, Apple, Commodore, CP/M systems running MBASIC, and many others. If you type it as you see it in the listing, it should work.

Some users may prefer to have the program run in a continuous loop so a number of geographic coordinates can be converted to QTH locators in one session. To make the program do that, add this line:

```
405 GOTO 80
```

Timex/Sinclair 1000 (ZX-81) users must make one modification to this program. On a Timex, an INPUT statement can't be combined with a "prompt" (i.e., a statement in quotation marks that tells the operator what data to enter). Instead, the prompt may be provided in a PRINT statement before the input.

For example, Timex users should modify line 90 and add a new line 92 as follows:

```
90 PRINT:PRINT "    LONGITUDE"
92 INPUT LO
```

The same thing must be done with all other input statements, which are found in lines 90, 100, 110, 120, and 130. Given that change, the program runs properly on a Timex with a 16K memory expansion module.

Operating notes

A program such as the GRIDLOCATOR is easy to use. Just enter the latitude and longitude in whole degrees, and the proper grid locator will be displayed on the screen.

Be sure to round *downward* (i.e., toward zero) to whole numbers. Even if you live at 33 degrees, 59.9 minutes north latitude, you fall into the 33-degree grid block, not the one that starts at 34 degrees. You also round toward zero for east longitude and south latitude coordinates, but you also add a minus sign ("–") before the coordinates.

Because 18 grid blocks converge at each pole, 90 degrees or –90 degrees are not treated as valid latitudes. If you plan an expedition to the North Pole, you would be rare DX indeed if you walk around the pole at 89.999 degrees north latitude, stepping from one rare grid block to the next. Or you could install separate antennas in each of the 18 grid blocks, perch yourself right on the pole, and use an 18-position rotary switch to transmit and receive from each of 18 different grid blocks... . The possibilities are endless.

But if you do that and you take along a computer, just enter the correct longitude for each point as you circle the pole, and use 89 for the latitude. The program will tell you which of the 18 blocks that converge at the pole you happen to be in at any moment.

Modifications for CP/M

None required.

GRIDLOCATOR (standard version)

```
10 REM GRIDLOCATOR PROGRAM:  CONVERTS LAT-LONG DATA TO
20 REM VHF-UHF LOCATORS ANYWHERE EXCEPT POLES. (STANDARD VERSION)
30 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
40 REM PUT YOUR CLEAR-SCREEN COMMAND HERE
50 PRINT:PRINT TAB(4)"GRIDLOCATOR CONVERSION PROGRAM"
60 PRINT:PRINT "ENTER LATITUDE AND LONGITUDE, USING '-' FOR S LAT, E LONG"
70 PRINT "   NOTE: ROUND DOWNWARD TO WHOLE NUMBERS"
80 DL$="":EQ$="":GM$=""
85 INPUT "   LATITUDE ";LA
90 PRINT:INPUT "   LONGITUDE ";LO
100 IF LO=180 THEN INPUT "EAST OR WEST OF DATE LINE (E/W)";DL$
110 IF LO=0 THEN INPUT "EAST OR WEST OF GREENWICH (E/W)";GM$
130 IF LA=0 THEN INPUT "NORTH OR SOUTH OF EQUATOR (N/S)";EQ$
```

```
140 IF LA=90 OR LA=-90 THEN GOSUB 450:END
150 LA=INT(LA):LO=INT(LO):L1=ABS(LO):L2=ABS(LA)
160 IF LO<0 THEN 180
170 A$=CHR$(73-INT(LO/20)):GOTO 190
180 A$=CHR$(74+INT(L1/20))
190 IF DL$="E" THEN A$="A"
200 IF DL$="W" THEN A$="R"
210 IF GM$="E" THEN A$="J"
220 IF GM$="W" THEN A$="I"
230 IF LA<0 THEN 250
240 B$=CHR$(74+INT(LA/10)):GOTO 260
250 B$=CHR$(73-INT(L2/10))
260 IF EQ$="N" THEN B$="J"
270 IF EQ$="S" THEN B$="I"
280 IF LO<0 THEN 300
290 C=9-INT(INT((LO/20-INT(LO/20))*20+.05)/2):GOTO 310
300 C=INT(INT((L1/20-INT(L1/20))*20+.05)/2)
310 IF DL$="W" OR GM$="W" THEN C=9
320 IF DL$="E" OR GM$="E" THEN C=0
330 IF LA<0 THEN 350
340 D=INT((LA/10-INT(LA/10))*10+.05):GOTO 360
350 D=9-INT((L2/10-INT(L2/10))*10+.05)
360 IF EQ$="N" THEN D=0
370 IF EQ$="S" THEN D=9
380 C=INT(C*10+.1)/10:D=INT(D*10+.1)/10
390 GR$=A$+B$+STR$(C)+STR$(D)
400 PRINT:PRINT "   GRID LOCATOR: ";GR$
420 PRINT:INPUT "WANT ANOTHER COMPUTATION (Y/N)";R$
430 IF R$="Y" THEN 80
440 END
450 PRINT "18 GRID BLOCKS CONVERGE HERE.  CHOOSE ONE":RETURN
```

CHAPTER

7

LATITUDE-LONGITUDE PROGRAMS FOR DX WORK

In a sense, almost every program in this book is for the DXer, that dedicated radio amateur who thinks nothing of staying up until all hours in pursuit of what may be only a momentary contact with a distant foreign station.

Because most DXers are interested in awards such as DXCC and WAZ, the AWARDS DATA BASE program (presented in the last chapter) is of special interest to them. And thousands of DXers participate in the many contests held each year, which means the CONTEST LOGGER and DUPECHECKER programs are programs for DXers. In addition, every DXer has to be concerned about putting out a solid signal, which means the antenna programs are also programs for DXers.

But in this chapter, we offer a package of programs that provide specialized information for the DXer—or for the casual operator who doesn't mind working a little exotic DX if it happens to come his way.

Most of the programs in this chapter are built around a large data file, a tabulation of more than 400 countries and cities around the world, with radio prefixes and latitude-longitude coordinates for each. This data base is shown on page 126. Once you have it safely stored on a floppy disk (or written into each program if you aren't using disk drives), there are a number of very useful things you can do.

You can, of course, produce customized beam heading and distance charts, charts that show the distance and direction from your home (or anyone else's) to places all over the world. You can also produce a chart that shows the sunrise and sunset times for all of these cities and

countries on any given day—a very important thing to know, especially if you're interested in DXing on 40 meters, 80 meters, or "top band," 160 meters.

Taking the sunrise chart idea a step further, you can also use the latitude-longitude data file to keep track of the "gray line." That is, you can determine which places around the world have sunrise and/or sunset times that closely coincide with yours. If you work DX on the lower frequencies, you know what happens to signal strengths at sunrise and sunset (especially when it's sunrise or sunset at both ends of the path).

In addition to the programs based on the latitude-longitude data file, there are other useful programs for DXers in this chapter. One short program uses the same mathematical formulas as the beam-heading and distance program, but without the data-handling and tabular printout features. It's ideal for quickly determining the direction and distance between any two points on the globe. All you need to know to use it is the latitude and longitude at each of the two points.

There's also a program here that prints a chart (or just displays it on the screen if you prefer) of sunrise/sunset times at any one place for any number of days. This program is ideal for the ham who is interested in contacting a specific place on a regular basis, or who is prepared to wait for the right day to make a difficult contact.

The Latitude-Longitude Data Base

The LATITUDE-LONGITUDE DATA BASE isn't a program in the usual sense. Rather, it's a collection of data that can be used by other programs to do a variety of different things. But because it's so central to the programs in the chapter, we'll describe it first.

The LATLONG file—as it is identified in the programs—is used as a sequential data file. Several programs read in its data and process it in various ways.

Referring to the full listing of the LATLONG data file, you'll notice that the text matter (called string variables when used in a program) appears in quotation marks. This isn't required on some computers, but it is on others, so we've reproduced it in that form. And you'll notice that the file ends with a strange-looking line that reads, "ZZZ","ZZZ",0,0. That is another feature not required on some computers, but it is needed to mark the end of the file on others. Some versions of BASIC have no other reliable way to detect the end of the data file without the program crashing.

The LATLONG file should be typed with a text editor or word processor such as Pie Writer on an Apple II, Wordstar on a CP/M computer, or Scripsit on a TRS-80. Any program capable of handling straight text on a line-by-line basis will do the job.

Because the performance of so many programs depends on its accuracy, the LATLONG file should be typed and proofread carefully. The programs in this chapter are only as good as the data file on which they are based. If the LATLONG file ends up with uncorrected typing errors,

some of the beam headings, distances, sunrise times, and gray line alignments will inevitably be incorrect.

Moreover, political boundaries, call signs, and place names are constantly changing. The LATLONG file was correct when it was prepared, but the data in it will inevitably become dated. If you're an avid DXer, you'll want to go through the data, deleting any countries that may have been deleted from the official ARRL countries list and entering any new ones that have been added. Also, when call prefixes are changed, the entries should be corrected and moved so the entire list remains in alphabetical order. If this file is kept up to date, it will provide useful service for many years.

Also, you may not want to retain all of the entries we included. For instance, if you live in the United States and your primary interest is worldwide DXing, you may prefer to omit the listing of domestic U.S. cities. Or you may want to check an atlas for coordinates and add specific cities of your own choosing.

If you revise the list, there are two things you'll need to do. First, put the new data into exactly the same form as the existing entries, using first the latitude and then the longitude in decimal degrees, and with east longitude and south latitude shown as negative numbers. Second, you should be sure the data file ends with the "ZZZ" entry, regardless of how many entries you insert ahead of it.

To derive decimal degrees, you could simply round off each coordinate to whole degrees. But for greater accuracy, the minutes can be converted into a decimal by dividing the number of minutes by 60. Add this decimal amount to the number of whole degrees. For example, 33 degrees, 45 minutes, equals 33.75 degrees. Don't forget the minus sign for south latitudes and east longitudes.

One other thing to remember if you modify the data file very much is that the programs are all written to handle a LATLONG data file with a maximum of 500 entries. As shown, the data file has 414 entries. If you go over 500, the DIM statements near the beginning of each program must be changed to accommodate the additional data lines. And one program, the DX DISPLAY, has a routine in it that will have to be modified if the number of lines is changed by more than a few. That will be discussed in the section on the DX DISPLAY program.

For VHF DXing, here's a suggestion: work up a version of the LATLONG data file for major cities and other locations of interest within, say, 1000 or 2000 miles of your own location, using QTH locators (see Chapter 6) in place of call-sign prefixes. Then you can use the BEAMHEADING program to produce a customized list of distances and beamheadings for various cities that you may work over long-haul tropo, meteor or E-skip paths on the VHF-UHF bands.

Using the LATLONG Data File without Disk Drives— Although the LATLONG data file is intended to be used on a computer with at least one disk drive, it can be used without that luxury. Some computers, such as the Commodore 64 and VIC-20, have provisions for loading a data file from a cassette. But there is also another way to read in

LATLONG data that will work on almost any computer: write the entire data file into each program as a series of READ and DATA statements.

In essence, here is how that's done. In the programs that rely on the data file, delete the OPEN, CLOSE, READ, and INPUT lines that refer to the LATLONG data file. The programs all have an INPUT line that enters the data one line at a time. Depending on the program (and the particular version of the program), it reads something like this: "INPUT#1,PX$(J),Q$(J),AX(J),OX(J)". Replace that with this line:

```
READ PX$(J),Q$(J),AX(J),OX(J)
```

If the particular program doesn't use the subscripted variables (the "(J)" part), don't use it in the read statement either. Then, at the end of the program, add the entire LATLONG data file by using a series of lines like these examples:

```
1000 DATA "A2","BOTSWANA",-22,-24
1010 DATA "A3","TONGA IS",-20,175
1020 DATA "A4X","OMAN",22,-58
```

and so forth, until the entire file is there. With the exception of the Timex/Sinclair (which doesn't have any provision for READ and DATA statements), this will work on any cassette-based computer with at least 24K of free memory.

You can also include this data in a number of different programs without retyping it every time. All you have to do is enter (and proofread) it once. Save that program on cassette, and then delete all of the program from memory except for the list of data statements. Then type another program ahead of the data lines, and save the whole thing again under its new file name.

LATLONG DATA FILE (standard version)

```
"A2","BOTSWANA",-22,-24
"A3","TONGA IS",-20,175
"A4X","OMAN",22,-58
"A5","BHUTAN",27.5,-90.1
"A6X","UNITED ARAB EMIRATES",24,-56
"A7X","QATAR",25,-51.2
"A9X","BAHRAIN",26,-50.5
"AP","PAKISTAN/KARACHI",25,-67
"BV","TAIWAN",23.5,-121
"BY","CHINA/PEKING",39.9,-116.4
"C2","NAURU",-.5,-167.9
"C3","ANDORRA",42.5,-1.5
"C5","THE GAMBIA",13.4,16
"C6","BAHAMAS/NASSAU",25,77.5
"C9","MOZAMBIQUE",-15,-41
"CE","CHILE/SANTIAGO",-33.5,70.8
"CE0A","EASTER IS",-28,109
"CE0X","SAN FELIX",-26,80
"CE0Z","JUAN FERNANDEZ",-34,78
"CO","CUBA/HAVANA",23.1,82.4
"CN","MOROCCO/CASABLANCA",33,8
"CP","BOLIVIA/LA PAZ",-16.5,68.4
```

```
"CR9","MACAO",22.15,-113.6
"CT","PORTUGAL/LISBON",38.7,9.2
"CT2","AZORES IS",38,25
"CT3","MADEIRA IS",33,17
"CX","URUGUAY/MONTEVIDEO",-34.9,56.2
"D2/3","ANGOLA",-13,-16
"D4","CAPE VERDE IS",16,24
"D6","COMOROS IS",-11.8,-43.7
"DA-DL","WEST GERMANY/BONN",51,-6
"DU","PHILIPPINES/MANILA",14.6,-121
"EA","SPAIN/MADRID",40.4,3.7
"EA6","BALEARIC IS",39.5,-3
"EA8","CANARY IS",28,15.5
"EA9","MELILLA & CEUTA",36.7,5.4
"EI","IRELAND/DUBLIN",53.3,6.3
"EL","LIBERIA/MONROVIA",7,11
"EP","IRAN/TEHRAN",35.8,-51.8
"ET","ETHIOPIA/ADDIS ABABA",9,-39
"F","FRANCE/PARIS",48.8,-2.3
"FB8W","CROZET IS",-46,-52
"FB8X","KERGUELEN IS",-50,-70
"FB8Y","ANTARCTICA/DUMONT D'URVILLE",-67,-140
"FB8Z","AMSTERDAM IS",-37.8,-77.5
"FC","CORSICA",42,-9
"FG","GUADELOUPE",17,62
"FH","MAYOTTE",-13,-45.3
"FK","NEW CALEDONIA",-22.3,-166.5
"FM","MARTINIQUE",14.7,61
"FO","CLIPPERTON",11,110
"FO","FR POLYNESIA/TAHITI",-18,150
"FP","ST PIERRE & MIQUELON IS",47,56.5
"FR","JUAN DE NOVA",-21.5,-40
"FR","GLORIOSO",-10.6,-47.3
"FR","REUNION",-21.1,-55.6
"FR","TROMELIN IS",-15.9,-52.2
"FS","ST MARTIN",18.1,63.1
"FW","WALLIS & FUTUNA",-14,176
"FY","FRENCH GUIANA",5,52
"G","ENGLAND/LONDON",51.5,0.1
"GD","ISLE OF MAN",54.2,4.5
"GI","NORTHERN IRELAND",54.6,5.9
"GJ/GC","JERSEY",49.3,2.2
"GM","SCOTLAND/GLASGOW",55.8,4.3
"GU/GC","GUERNSEY",49.5,2.7
"GW","WALES",52.5,3.5
"H4/VR4","SOLOMON IS",-10,-159
"HA","HUNGARY/BUDAPEST",47.5,-19.1
"HB","SWITZERLAND",47,-7
"HB0","LIECHTENSTEIN",47,-9.5
"HC","ECUADOR/QUITO",-.2,78
"HC8","GALAPAGOS IS",-2,92
"HH","HAITI",19,72.4
"HI","DOMINICAN REPUBLIC",18.5,70
"HK","COLUMBIA/BOGOTA",5,74
"HK0","BAJO NUEVO",15.9,78.7
"HK0","MALPELO",4,81.6
"HK0","SAN ANDRES",12.5,81.7
"HL/HM","KOREA/SEOUL",37.5,-127
"HP","PANAMA",9,80
"HR","HONDURAS",14,87
"HS","THAILAND/BANGKOK",13.8,-100.5
"HV","VATICAN CITY",41.9,-12.5
"HZ/7Z","SAUDI ARABIA/RIYADH",24.5,-47
"I","ITALY/ROME",41.9,12.5
"IS","SARDINIA",40,-9
"J2/FL8","DJIBOUTI",11.6,-43.2
"J3/VP2G","GRENADA",12.1,61.7
"J5","GUINEA-BISSAU",12,15
"J6","ST. LUCIA",14,61
"J7","DOMINICA",15.4,61.3
```

```
"JA","JAPAN/TOKYO",35.7,-139.8
"JA","JAPAN/OSAKA",34.8,-135.8
"JA","JAPAN/SAPPORO",43,-141
"JD","OGASAWARA",27.5,-141
"JT","MINAMI-TORISHIMA",25,-154
"JW","MONGOLIA/ULAN BATOR",48,-107
"JX","SVALBARD IS",78,-16
"JY","JORDAN/AMMAN",32,-36
"KC4","ANTARCTICA/BYRD STN",-80,120
"KC4","ANTARCTICA/MCMURDO STN",-77.5,-167
"KC6","E CAROLINE IS",8,-156
"KC6","W CAROLINE IS",9.5,-138.2
"KG4","GUANTANAMO BAY",20,75
"KH1","BAKER & PHOENIX",0.3,176.5
"KH2","GUAM",14,-145
"KH3","JOHNSTON IS",17,170
"KH4","MIDWAY IS",28.2,177.25
"KH5","PALMYRA IS",6,162
"KH5K","KINGMAN REEF",7.5,162.8
"KH6","HAWAII/HONOLULU",21.3,157.9
"KH7","KURE IS",28.4,178.8
"KH8","AMERICAN SAMOA",-14,171
"KH9","WAKE IS",19.3,-167
"KH0","MARIANA IS/SAIPAN",16,-146
"KL7","ALASKA/ANCHORAGE",61.2,150
"KP1","NAVASSA IS",18.4,75
"KP2","VIRGIN IS",18.3,64.7
"KP4","PUERTO RICO/SAN JUAN",18.5,66.2
"KP4","DESECHEO",18.3,67.5
"KS4/HK0","SERRANA BANK",14.4,80.2
"KX","MARSHALL IS",12,-167
"LA","NORWAY/OSLO",60,-10.7
"LU","ARGENTINA/BUENOS AIRES",-34,58.4
"LX","LUXEMBOURG",49,-6.5
"LZ","BULGARIA/SOFIA",42.7,-23.3
"OA","PERU/LIMA",-12,77
"OD","LEBANON/BEIRUT",33.9,-35.5
"OE","AUSTRIA/VIENNA",48.2,-16.3
"OH","FINLAND/HELSINKI",60.2,-25
"OH0","ALAND IS",60.25,-20
"OJ0","MARKET REEF",60.3,-19
"OK","CZECHOSLOVAKIA/PRAGUE",50.1,-14.4
"ON","BELGIUM/BRUSSELS",50.9,-4.4
"OX/XP","GREENLAND/GODTHAAB",63,52
"OY","FAEROE IS",62,7
"OZ","DENMARK/COPENHAGEN",55.7,-12.6
"P2","PAPUA NEW GUINEA",-9.4,-147.1
"PA","NETHERLANDS/AMSTERDAM",52.4,-4.9
"PJ","NETH. ANTILLES",12,69
"PJ7/8","ST MAARTEN",18,63
"PY","BRAZIL/RIO DE JANEIRO",-23,43.2
"PY0","FERNANDO DE NORONHA",-3,32
"PY0","ST PETER-PAUL'S ROCKS",1.5,30
"PY0","TRINIDADE IS",19,29
"PZ","SURINAM",6,55
"S2","BANGLADESH/DACCA",24,-90.5
"S7","SEYCHELLES IS",-4,-56
"S9","SAO TOME",.3,-6
"SM","SWEDEN/STOCKHOLM",59.3,-18.1
"SP","POLAND/WARSAW",52.2,-21
"ST","SUDAN/KHARTOUM",15.5,-32.5
"ST0",SOUTHERN SUDAN,5,-31.6
"SU","EGYPT/CAIRO",30,-31.4
"SV","GREECE/ATHENS",38,-23.7
"SV","CRETE",35.5,-25.2
"SV","DODECANESE IS",36.5,-27.5
"SV","MT ATHOS",40.2,-24.3
"T2","TUVALU",-8,-178
```

```
"T3","CENTRAL KIRIBATI",-4,171
"T3","EAST KIRIBATI",-10,150
"T3","WEST KIRIBATI",-5,-175
"TA","TURKEY/ANKARA",39.9,-32.9
"TF","ICELAND/REYKJAVIK",64.1,22
"TG","GUATEMALA",14.6,90.5
"TI","COSTA RICA",10,84
"TI9","COCOS IS",5,87.5
"TJ","CAMEROON",4,-12
"TL","CENTRAL AFR REP/BANGUI",4.5,-18
"TN","CONGO/BRAZZAVILLE",-4,-15
"TR","GABON REP",-.5,-9
"TT","CHAD REP",12,-15
"TU","IVORY COAST",5.5,4
"TY","BENIN",6.3,-2.3
"TZ","MALI REP",13,8
"UA1","EUR USSR/LENINGRAD",60,-30
"UA3","EUR USSR/MOSCOW",56,-37.5
"UA4","EUR USSR/KUYBYSHEV",53.5,-50.5
"UA6","EUR USSR/ROSTOV",47,-40
"UA9","ASIATIC USSR/PERM",57.5,-56
"UA0","AS.USSR/KRASNOYARSK",56,-93
"UA0","AS.USSR/KHABAROVSK",48,-135
"UA1","FRANZ JOSEF LAND",80,-50
"UA2","KALININGRAD",55,-20.5
"UB5","UKRAINE/KIEV",50.5,-31
"UC2","WHITE RUSSIA/MINSK",54,-27.5
"UD6","AZERBAIJAN/BAKU",40.5,-50
"UF6","GEORGIA/TBILISI",41.5,-45
"UG6","ARMENIA/ERIVAN",40.5,-44.5
"UH8","TURKOMAN/ASHKHABAD",38,-58.0
"UI8","UZBEK/BUKHARA",40,-64
"UJ8","TADZHIK/SAMARKAND",39.5,-67
"UL7","KAZAKH/ALMA-ATA",53,-76
"UM8","KIRGHIZ",42,-75
"UO5","MOLDAVIA/KISHINEV",47,-29
"UP2","LITHUANIA/VILNA",54.5,-25.5
"UQ2","LATVIA/RIGA",57,-24
"UR2","ESTONIA/TALLINN",59,-25
"VE1","SABLE IS",44,60
"VE1","ST PAUL IS",47.2,60.2
"VE1","CANADA/SAINT JOHN (NB)",45.3,66.1
"VE1","NS/HALIFAX",44.6,63.6
"VE2","PQ/MONTREAL",45.5,73.6
"VE3","ONT/TORONTO",43.7,79.4
"VE3","ONT/THUNDER BAY",48.4,89.2
"VE4","MAN/WINNIPEG",49.9,97.1
"VE5","SASK/REGINA",50.5,104.6
"VE6","ALTA/CALGARY",51,114.1
"VE6","ALTA/EDMONTON",53.5,113.5
"VE7","BC/VANCOUVER",49.3,123.1
"VE8","YUKON/WHITEHORSE",60.7,135.1
"VE8","NWT/YELLOWKNIFE",62.5,114.4
"VO1","NEWFOUNDLAND/ST.JOHN'S",47.6,52.7
"VO2","LABRADOR/GOOSE BAY",51.3,60.4
"VK1","AUSTRALIA/CANBERRA",-35.5,-149
"VK2","AUSTRALIA/SYDNEY",-34,-151
"VK3","AUSTRALIA/MELBOURNE",-38,-145
"VK4","AUSTRALIA/BRISBANE",-27.5,-153
"VK5","AUSTRALIA/ADELAIDE",-34.9,-138.6
"VK6","AUSTRALIA/PERTH",-32,-115.9
"VK7","AUSTRALIA/HOBART",-42.9,-147.3
"VK8","AUSTRALIA/DARWIN",-12.5,-130.9
"VK2","LORD HOWE IS",-31.6,-159.1
"VK9","WILLIS IS",-16.3,-150
"VK9","COCOS IS",-12,-97
"VK9","MELLISH REEF",-17.4,-155.85
"VK9","NORFOLK IS",-29,-167.95
"VK9X","CHRISTMAS IS",-10.5,-105.65
"VK0","HEARD IS",-53,-75
```

```
"VK0","MACQUARIE IS",-54.6,-158.9
"VP1","BELIZE",17.5,88.3
"VP2A","ANTIGUA",17,59.8
"VP2E","ANGUILLA",18.3,63
"VP2K","ST KITTS",17.3,62.6
"VP2M","MONTSERRAT",16.8,62.2
"VP2S","ST VINCENT",13.3,61.3
"VP2V","BRITISH VIRGIN IS",18.4,64.6
"VP5","TURKS & CAICOS IS",22,72
"VP8","FALKLAND IS",-52,60
"VP8","S GEORGIA IS",-54,37
"VP8","S ORKNEY IS",-60,46
"VP8","S SANDWICH IS",-57,28
"VP8","S SHETLAND IS",-62,58.5
"VP9","BERMUDA",32.3,64.7
"VQ9","CHAGOS/DIEGO GARCIA",-7.3,-72.4
"VR6","PITCAIRN IS",-25.1,130.1
"VS5","BRUNEI",5,-115
"VS6","HONG KONG",22.5,-114
"VU","INDIA/BOMBAY",19,-72.8
"VU","INDIA/CALCUTTA",22.6,-88.4
"VU","INDIA/NEW DELHI",28,-77
"VU7","ANDAMAN & NICOBAR IS",10,-94
"VU7","LACCADIVE IS",10,-70
"W1","CONN/HARTFORD",41.8,72.7
"W1","MAINE/BANGOR",44.8,68.8
"W1","MASS/BOSTON",42.4,71.1
"W1","NH/CONCORD",43.2,71.5
"W1","RI/PROVIDENCE",41.8,71.4
"W1","VT/MONTPELIER",44.3,72.6
"W2","NJ/TRENTON",40.1,74.8
"W2","NY/NEW YORK CITY",40.8,74
"W2","NY/BUFFALO",42.9,78.9
"W3","DEL/WILMINGTON",39.7,75.5
"W3","MD/BALTIMORE",39.3,76.6
"W3","PENN/PHILADELPHIA",40,75.2
"W3","PENN/PITTSBURGH",40.4,80
"W3","WASHINGTON DC",38.9,77
"W4","ALA/MONTGOMERY",32.4,86.3
"W4","FLA/MIAMI",25.7,80.3
"W4","GA/ATLANTA",33.7,84.4
"W4","KY/LOUISVILLE",38.2,85.8
"W4","NC/RALEIGH",35.8,78.7
"W4","SC/COLUMBIA",34,81
"W4","TENN/KNOXVILLE",35.9,84
"W4","TENN/MEMPHIS",35.1,90.1
"W4","VA/RICHMOND",37.5,77.5
"W5","ARK/LITTLE ROCK",34.7,92.4
"W5","LA/NEW ORLEANS",30,90.1
"W5","MISS/JACKSON",32.3,90.3
"W5","NM/ALBUQUERQUE",35.2,106.7
"W5","OKLA/OKLA CITY",35.5,97.6
"W5","TEXAS/DALLAS",32.8,96.8
"W5","TEXAS/HOUSTON",29.7,95.4
"W5","TEXAS/SAN ANTONIO",29.4,98.5
"W6","CAL/LOS ANGELES",34.1,118.2
"W6","CAL/SAN DIEGO",32.7,117.2
"W6","CAL/SAN FRANCISCO",37.7,121.5
"W7","ARIZ/PHOENIX",33.5,112.1
"W7","IDAHO/BOISE",43.6,116.2
"W7","MONT/HELENA",46.6,112
"W7","NEV/LAS VEGAS",36.2,115.2
"W7","OREG/PORTLAND",45.5,122.7
"W7","UTAH/SALT LAKE",40.8,111.9
"W7","WASH/SEATTLE",47.5,122.4
"W7","WASH/SPOKANE",47.6,117.5
"W7","WYO/CHEYENNE",41.2,104.8
"W8","MICH/DETROIT",42.4,83.1
"W8","OHIO/CINCINNATI",39.2,84.6
"W8","OHIO/CLEVELAND",41.5,81.7
```

```
"W8","WVA/CHARLESTON",38.4,81.7
"W9","ILL/CHICAGO",41.8,87.7
"W9","IND/INDIANAPOLIS",39.7,86.2
"W9","WISC/MILWAUKEE",43,87.9
"WO","COLO/DENVER",39.7,104.9
"WO","IOWA/DES MOINES",41.6,93.6
"WO","KANSAS/WICHITA",37.7,97.4
"WO","MINN/MINNEAPOLIS",45,93.3
"WO","MO/KANSAS CITY",39.1,94.6
"WO","MO/ST. LOUIS",38.7,90.3
"WO","NEBR/OMAHA",41.3,96
"WO","ND/BISMARCK",46.8,100.8
"WO","SD/PIERRE",44.4,100.4
"XE","MEXICO/MEXICO CITY",19.4,99.1
"XF4","REVILLA GIGEDO",19,111.5
"XT","UPPER VOLTA",12.3,1.7
"XU","KHMER REP/PHNOM PENH",11.5,-105
"XV","VIETNAM/HO CHI MINH CITY",10.8,-106.7
"XW","LAO REP/VIENTIANE",18,-102.5
"XZ","BURMA/RANGOON",16.8,-96
"Y2-9","EAST GERMANY/BERLIN",52.5,-13.4
"YA","AFGHANISTAN/KABUL",34,-68.5
"YB/YC","INDONESIA/JAKARTA",-6,-107
"YI","IRAQ/BAGHDAD",33,-44.5
"YJ","NEW HEBRIDES",-18,-168
"YK","SYRIA/DAMASCUS",34,-36.5
"YN","NICARAGUA",12,86
"YO","ROMANIA/BUCHAREST",44.4,-26.1
"YS","EL SALVADOR",13.8,88.9
"YU","YUGOSLAVIA/BELGRADE",45,-20.5
"YV","VENEZUELA/CARACAS",10.5,67
"YV0","AVES IS",12,67.5
"ZA","ALBANIA/TIRANA",41.5,-19.5
"ZB","GIBRALTAR",36.2,5.5
"ZD7","SAINT HELENA",-16,6
"ZD8","ASCENSION IS",-7.9,14.3
"ZD9","TRISTAN DA CUNHA",-38,20
"ZE","ZIMBABWE/SALISBURY",-18,-31
"ZF","CAYMAN IS",19.5,81.2
"ZK1","S COOK IS",-21,159.5
"ZK1","N COOK IS",-10.5,161.5
"ZK2","NIUE",-19,170
"ZL","AUCKLAND & CAMPBELL IS",-52.5,-169
"ZL","CHATHAM IS",-44,176
"ZL","KERMADEC IS",-30,178
"ZL1","NEW ZEALAND/AUCKLAND",-36.9,-174.8
"ZL2","NZ/WELLINGTON",-41.3,-174.8
"ZL3","NZ/CHRISTCHURCH",-43,-172.5
"ZL4","NZ/DUNEDIN",-46,-170
"ZL5","ANTARCTICA/SCOTT BASE",-78,-165
"ZM7","TOKELAUS",-11,165
"ZP","PARAGUAY/ASUNCION",-25.3,58
"ZS1","SO AFR/CAPETOWN",-33.9,-18.4
"ZS2","MARION & P.E. IS",-46.8,-37.8
"ZS3","NAMIBIA (SW AFRICA)",-22.5,-17
"ZS5","SO AFR/DURBAN",-29.9,-30.9
"ZS6","SO AFR/JOHANNESBURG",-26.2,-28.1
"1S","SPRATLY",8.8,-111.9
"3A","MONACO",43.7,-7.4
"3B6/7","ST BRANDON & AGALEGA",-10.4,-56.6
"3B8","MAURITIUS",-20.3,-57.5
"3B9","RODRIGUES",-19.7,-63.4
"3C","EQUATORIAL GUINEA",2,-9
"3C0","ANNOBON",3.5,-8.5
"3D2","FIJI IS",-17,-178
"3D6","SWAZILAND",-27,-32
"3V","TUNISIA/TUNIS",36.8,-10
"3X","GUINEA/CONAKRY",10,13
"3Y","BOUVET IS",-54,-3.4
"4S","SRI LANKA (CEYLON)",7,-79.9
```

```
"4U","GENEVA (ITU)",46.2,-6.2
"4U","UN HQ NEW YORK",40.7,74
"4W","YEMEN",15,-44
"4X/4Z","ISRAEL",32,-35
"5A","LIBYA/TRIPOLI",32.5,-13
"5B/ZC","CYPRUS",35,-33
"5H","TANZANIA/DAR ES SALAAM",-7,-39.5
"5N","NIGERIA",6.5,-3
"5R","MALAGASY REP",-18.5,-47.5
"5T","MAURITANIA",18,16
"5U","NIGER REP",13.5,-2
"5V","TOGO REP",6,-1.5
"5W","WESTERN SAMOA",-13,172
"5X","UGANDA/KAMPALA",1,-32
"5Z","KENYA/NAIROBI",-1.5,-38
"6O","SOMALI/MOGADISHO",2,-46
"6W","SENEGAL/DAKAR",15,18
"6Y","JAMAICA",18,76
"7O","S YEMEN",13,-45.2
"7P","LESOTHO",-29,-28
"7Q","MALAWI",-15,-35
"7X","ALGERIA/ALGIERS",36.7,-3
"8P","BARBADOS",11.5,59.5
"8Q","MALDIVE IS",.5,-73
"8R","GUYANA",6.8,58.2
"8Z4","SAUDI ARABIA/IRAQ N.Z.",28,-47
"9A/M1","SAN MARINO",44,-13
"9G","GHANA/ACCRA",6,1
"9H","MALTA",36,-14.4
"9J","ZAMBIA/LUSAKA",-15,-28
"9K","KUWAIT",29.5,-47.8
"9L","SIERRA LEONE",8,13.2
"9M2","W MALAYSIA",3,-102
"9M6","E MALAYSIA/SABAH",5,-117
"9M8","E MALAYSIA/SARAWAK",2,-113
"9N","NEPAL",27.5,-85
"9Q","ZAIRE/KINSHASA",-5,-15.3
"9U","BURUNDI",-3,-29
"9V","SINGAPORE",1.3,-103.8
"9X","RWANDA",-1.5,-30
"9Y","TRINIDAD",11,62
" - ","ABU AIL",27.5,-50
"ZZZ","ZZZ",0,0
```

THE BEAMHEADING AND DISTANCE PROGRAM

What it does

The BEAMHEADING CHART program, as we indicated earlier, will prepare beamheading and distance tables that are customized for any location.

The program uses the LATLONG data file as the basis for its computations. The first page of a sample beamheading chart is shown here. The chart is as long or as short as the LATLONG data file; if you want a shorter version of the chart with fewer countries, make a scaled-down version of the LATLONG data file. And as we said earlier, VHF DXers can use the program with a localized version of the LATLONG data file to produce beamheading and distance charts for tropo, meteor scatter, and E-skip work.

Setup instructions

As listed in the various versions, the program runs correctly on a variety of computers. Few changes are necessary for most applications. Those wishing to use the program for accurate close-in computations (for

Sample printout of the BEAMHEADING CHART program.

BEAM HEADING AND DISTANCE PROGRAM
FOR DXCC COUNTRIES AND MAJOR CITIES

BEAM HEADINGS AND DISTANCES CUSTOMIZED FOR N6NB
LOCATED AT 33 - 50 NORTH LATITUDE AND 117 - 45 WEST LONGITUDE

ALL MEASUREMENTS ARE TO THE CENTER OF THE COUNTRY WHEN NO CITY IS SPECIFIED. HEADINGS ARE IN DEGREES EAST OF NORTH, AND DISTANCES ARE IN STATUTE MILES VIA SHORT PATH.

PREFIX	LOCATION	BEARING	RANGE
A2	BOTSWANA	81	9,972
A3	TONGA IS	234	5,289
A4X	OMAN	5	8,563
A5	BHUTAN	333	7,818
A6X	UNITED ARAB EMIRATES	7	8,414
A7X	QATAR	12	8,302
A9X	BAHRAIN	12	8,226
AP	PAKISTAN,KARACHI	355	8,354
BV	TAIWAN	307	6,903
BY	CHINA,PEKING	322	6,278
C2	NAURU	261	5,340
C3	ANDORRA	40	5,910
C5	THE GAMBIA	72	6,355
C6	BAHAMAS,NASSAU	93	2,480
C9	MOZAMBIQUE	51	10,572
CE	CHILE,SANTIAGO	142	5,556
CE0A	EASTER IS	171	4,308
CE0X	SAN FELIX	144	4,815
CE0Z	JUAN FERNANDEZ	147	5,344
CO	CUBA,HAVANA	100	2,257
CN	MOROCCO,CASABLANCA	52	5,946
CP	BOLIVIA,LA PAZ	129	4,754
CR9	MACAO	311	7,298
CT	PORTUGAL,LISBON	48	5,651
CT2	AZORES IS	56	4,962
CT3	MADEIRA IS	57	5,525
CX	URUGUAY,MONTEVIDEO	134	6,191
D2,3	ANGOLA	75	9,199
D4	CAPE VERDE IS	75	5,813
D6	COMOROS IS	42	10,510
DA-DL	WEST GERMANY,BONN	32	5,649
DU	PHILIPPINES,MANILA	301	7,323
EA	SPAIN,MADRID	44	5,806
EA6	BALEARIC IS	42	6,110
EA8	CANARY IS	60	5,795
EA9	MELILLA & CEUTA	48	5,899
EI	IRELAND,DUBLIN	35	5,154
EL	LIBERIA,MONROVIA	75	6,889
EP	IRAN,TEHRAN	9	7,573
ET	ETHIOPIA,ADDIS ABABA	32	9,101
F	FRANCE,PARIS	35	5,639
FB8W	CROZET IS	150	11,430
FB8X	KERGUELEN IS	197	11,245
FB8Y	ANTARCTICA,DUMONT D'URVILLE	208	8,669
FB8Z	AMSTERDAM IS	248	11,533
FC	CORSICA	37	6,202
FG	GUADELOUPE	94	3,619
FH	MAYOTTE	41	10,639
FK	NEW CALEDONIA	244	6,302

VHF work, for instance) may prefer to eliminate the rounding of distances to whole miles in line 670 (line 650 in the Apple version). If that line is deleted, you will get mileage figures that are correct to a fraction of a mile, but the data may not tabulate correctly in the printouts. Also, the accuracy of the computations will still be limited by the accuracy of the coordinates that are entered in the first place. If the coordinates (both the ones in the data file and your local ones) are precisely correct, the result will reflect that precision.

The accuracy of this and other distance-calculating programs that use spherical trigonometry is limited by the inexact shape of the earth: our planet isn't perfectly round. However, this variable is a very minor one and can be ignored for most practical purposes. Even the Federal Communications Commission's computations of mandatory mileage separations for broadcast stations—computations in which millions of corporate dollars are at stake—are based on formulas that ignore the out-of-roundness of the earth. Computations using the programs in this chapter consistently produce answers within *feet* (not miles) of those obtained with the FCC's mileage separation formulas.

For most long-haul DX work, of course, that kind of down-to-the-decimal precision is rarely needed.

Operating notes

Few programs are simpler to use than the BEAMHEADING CHART program. To run it, you respond to a few questions at the beginning. You'll enter your call sign (so it can be printed out at the top of each customized chart), and then your longitude and latitude. For east longitude or south latitude, be sure to put a minus sign in front of both the degrees *and* minutes as you enter the values.

If you request a hard-copy printout, you'll have a custom beamheading chart in a few minutes. If you opt instead to merely see a display of the headings and distances, you can stop and restart the display by typing a control-S on most computers.

Modifications for CP/M

The TRS-80 and CP/M listing should be modified as follows for CP/M systems running Microsoft BASIC-80, version 5.0 or later:

1. Delete the CLEAR statement in line 35;
2. Replace the clear-screen command in line 30 with the proper command for your system;
3. In line 210, which sets up the format for later PRINT USING and LPRINT USING statements, replace all percent signs ("%") with a backslash ("\").

BEAMHEADING CHART (TRS-80 and CP/M)

```
10 REM ANTENNA HEADING AND DISTANCE PROGRAM (TRS-80 VERSION)
20 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
30 CLS:REM CLEAR SCREEN
35 CLEAR 250
40 B1$=" BEAM HEADING AND DISTANCE PROGRAM"
50 B2$="FOR DXCC COUNTRIES AND MAJOR CITIES"
60 PRINT TAB(14) B1$:PRINT TAB(14) B2$:PRINT:PRINT:PRINT
80 PRINT "PLEASE BE SURE THE DATA FILE (LATLONG) IS ON DISK"
90 PRINT
100 INPUT "WHAT IS YOUR CALL SIGN";CL$
110 PRINT
```

```
120 PRINT "ENTER COORDINATES, USING '-' FOR DEGREES AND MINS"
122 PRINT "   IF EAST LONGITUDE OR SOUTH LATITUDE"
140 INPUT "   ENTER YOUR LATITUDE IN DEGREES, MINUTES (E.G. 33,30)";LA,AM
160 INPUT "   ENTER YOUR LONGITUDE IN DEGREES, MINUTES (E.G.116,30)";LO,OM
170 PRINT: INPUT "DO YOU WANT A HARD COPY PRINTOUT (Y/N)";LP$
180 PRINT:PRINT
190 PI=3.14159
200 F$= "     PREFIX          LOCATION              BEARING        RANGE"
210 F1$="     %    %    %                          %   ###          ##,###"
220 IF LA<90 AND LA>-90 AND LO<180 AND LO>-180 GOTO 260
230 PRINT "NOTE:   LATITUDE MUST BE MORE THAN -90 AND LESS THAN +90."
240 PRINT "        LONGITUDE MUST BE MORE THAN -180 AND LESS THAN +180."
250 GOTO 140
260 IF LA=>0 THEN NO$="NORTH LATITUDE" ELSE NO$="SOUTH LATITUDE"
270 IF LO=>0 THEN WE$="WEST LONGITUDE" ELSE WE$="EAST LONGITUDE"
280 IF LP$<>"Y" THEN 390
290 FOR J=1 TO 5:LPRINT:NEXT J
300 LPRINT TAB(22) B1$:LPRINT TAB(22) B2$:LPRINT:LPRINT
310 LPRINT TAB(16) "BEAM HEADINGS AND DISTANCES CUSTOMIZED FOR ";CL$
320 LPRINT TAB(8) "LOCATED AT ";LA;"-";AM;NO$;" AND ";LO;"-";OM;WE$
330 LPRINT:LPRINT:LPRINT
340 LPRINT TAB(10) "ALL MEASUREMENTS ARE TO THE CENTER OF THE COUNTRY WHEN"
350 LPRINT TAB(10) "NO CITY IS SPECIFIED.  HEADINGS ARE IN DEGREES EAST OF"
360 LPRINT TAB(10) "NORTH, AND DISTANCES ARE IN STATUTE MILES VIA SHORT PATH."
370 LPRINT:LPRINT
380 LPRINT F$:LPRINT
390 RAD=57.2958
400 LT=(LA+AM/60)/RAD
410 LG=(LO+OM/60)/RAD
420 PRINT F$:PRINT
430 OPEN "I",1,"LATLONG"
440 LN=21
450 IF EOF(1) THEN 710
460 LN=LN+1
470 IF LN=60 AND LP$="Y" THEN GOSUB 750
480 INPUT #1, PX$,Q$,A1,O1
482 IF PX$="ZZZ" THEN 710
490 A1=A1/RAD
500 O1=O1/RAD
510 L=LG-O1:IF L=0 THEN 780
520 IF L>PI THEN L=L-2*PI
530 IF L<-PI THEN L=L+2*PI
540 AA=COS(L)*(COS(A1)/SIN(A1)):AA=ATN(AA)
550 AZ=((COS(L)/SIN(L))*COS(LT+AA))/SIN(AA):AZ=ATN(1/AZ)
560 AZ=AZ*RAD:L=L*RAD
570 IF L>0 AND AZ<0 THEN AZ=AZ+180
580 IF L<0 AND AZ>0 THEN AZ=AZ+180
590 IF L<0 AND AZ<0 THEN AZ=AZ+360
600 AZ=INT(AZ+.5)
610 DX=(SIN(LT)*SIN(A1))+(COS(LT)*COS(A1)*COS(L/RAD))
620 DX=SQR(1-(DX*DX))/DX
630 IF LG=O1 THEN 800
640 DX=ATN(DX)
650 DX=DX*RAD:IF DX<0 AND DX>-90 THEN DX=DX+180
660 DX=DX*69.0468
670 DX=INT(DX+.5)
680 PRINT USING F1$;PX$,Q$,AZ,DX
690 IF LP$="Y" THEN LPRINT USING F1$;PX$,Q$,AZ,DX
700 GOTO 450
710 IF LP$<>"Y" THEN 730
720 FOR J=LN TO 65:LPRINT:NEXT J
730 CLOSE
740 END
750 FOR J=1 TO 11:LPRINT:NEXT J
760 LPRINT F$:LPRINT:LN=7
770 RETURN
780 IF LT>A1 THEN AZ=180:GOTO 600
790 IF LT<A1 THEN AZ=0:GOTO 600
800 IF LT>A1 THEN DX=LT-A1:GOTO 650
810 IF LT<A1 THEN DX=A1-LT:GOTO 650
```

BEAMHEADING CHART (Apple)

```
 10  REM  ANTENNA HEADING AND DISTANCE PROGRAM (APPLE)
 15  REM   PRINTS OUT A CHART CUSTOMIZED FOR ANY LOCATION
 20  REM  BY N6NB AND KC6A; (C) 1983 BY HAYDEN PUBLISHING CO.,INC.
 30  HOME
 40 B1$ = " BEAM HEADING AND DISTANCE PROGRAM"
 50 B2$ = "FOR DXCC COUNTRIES AND MAJOR CITIES"
 60  PRINT "   "B1$: PRINT : PRINT "   "B2$: PRINT : PRINT
 80  FOR J = 1 TO 10: PRINT : NEXT J
 90  INPUT "WHAT IS YOUR CALL SIGN? ";CL$
100  PRINT : PRINT "IN ENTERING COORDINATES, USE '-' FOR"
102  PRINT "BOTH DEGREES AND MINUTES IF"
110  PRINT "SOUTH LATITUDE, EAST LONGITUDE"
120  PRINT
130  PRINT "WHAT IS YOUR LATITUDE": INPUT "IN DEGREES, MINUTES (E.G. 33,30)? ";LA,
AM
140  IF LA < 90 AND LA >  - 90 GOTO 160
150  PRINT "LATITUDE MUST BE BETWEEN -90 AND +90": GOTO 130
160  PRINT
170  PRINT "WHAT IS YOUR LONGITUDE"
180  INPUT "IN DEGREES, MINUTES (E.G. 85,15)? ";LO,OM
190  IF LO < 180 AND LO >  - 180 GOTO 210
200  PRINT "LONGITUDE MUST BE BETWEEN -180 AND +180": GOTO 180
210  PRINT
220 NO$ = " SOUTH LATITUDE"
230  IF LA > 0 THEN NO$ = " NORTH LATITUDE"
240 WE$ = " EAST LONGITUDE"
250  IF LO > 0 THEN WE$ = " WEST LONGITUDE"
260  INPUT "DO YOU WANT A HARD COPY PRINTOUT? (Y/N) ";LP$
290  PRINT : PRINT
300 D$ =  CHR$ (4):I$ =  CHR$ (9):P$ = "PR#0":P1$ = "PR#1":X$ = "80N"
320 F$ = "    PREFIX          LOCATION                    BEARING     RANGE"
330 RA = 57.2958:PI = 3.14159
350  PRINT "PLEASE NOTE:  LATLONG DATA FILE MUST BE ON DISK."
370  PRINT : PRINT : PRINT
378  IF LP$ = "Y" THEN  PRINT D$;P1$: PRINT I$;X$: GOSUB 940
380  IF LP$ <  > "Y" THEN  PRINT "TYPE CTRL-S TO STOP AND RESTART PROGRAM": PRINT
410  PRINT "PREFIX    LOCATION       BEARING  RANGE"
420  PRINT
430  PRINT D$;"OPEN LATLONG DATA"
440 LT = (LA + AM / 60) / RA
450 LG = (LO + OM / 60) / RA
460 LN = 19
470  PRINT D$;"READ LATLONG DATA"
480 LN = LN + 1
490  IF LN = 60 AND LP$ = "Y" THEN 840
500  INPUT PX$,Q$,A1,O1
510  IF PX$ = "ZZZ" THEN  GOTO 780
520 A1 = A1 / RA
530 O1 = O1 / RA
540 L = LG - O1: IF L = 0 THEN 1080
550  IF L > PI THEN L = L - 2 * PI
560  IF L <  - PI THEN L = L + 2 * PI
570 AA =  COS (L) * ( COS (A1) /  SIN (A1)):AA =  ATN (AA)
580 AZ = (( COS (L) /  SIN (L)) *  COS (LT + AA)) /  SIN (AA):AZ =  ATN (1 / AZ)
590 AZ = AZ * RA:L = L * RA
600  IF L > 0 AND AZ < 0 THEN AZ = AZ + 180
602  IF L < 0 AND AZ > 0 THEN AZ = AZ + 180
604  IF L < 0 AND AZ < 0 THEN AZ = AZ + 360.
606 AZ =  INT (AZ + .5)
610 DX = ( SIN (LT) *  SIN (A1)) + ( COS (LT) *  COS (A1) *  COS (L / RA))
612 DX =  SQR (1 - (DX * DX)) / DX
614  IF LG = O1 THEN 1084
620 DX =  ATN (DX)
630 DX = DX * RA: IF DX < 0 AND DX >  - 90 THEN DX = DX + 180
640 DX = DX * 69.0468
650 DX =  INT (DX + .5)
710 AZ$ =  STR$ (AZ)
```

```
720  IF AZ < 100 THEN AZ$ = "0" + AZ$
730  IF AZ < 10 THEN AZ$ = "0" + AZ$
750  IF LP$ = "Y" THEN  PRINT D$;P1$: GOSUB 890
760  GOSUB 920
770  GOTO 470
780  PRINT D$;"CLOSE"
790  IF LP$ <  > "Y" GOTO 830
800  PRINT D$;P1$
810  FOR J = LN TO 66: PRINT : NEXT J
820  PRINT D$;P$
830  END
840  PRINT D$;P1$
850  FOR J = 1 TO 11: PRINT : NEXT J
860  PRINT F$: PRINT :LN = 6
870  PRINT D$;P$
880  GOTO 470
890  PRINT I$;X$; SPC( 4);PX$; SPC( 9 -  LEN (PX$));Q$; SPC( 35 -  LEN (Q$));AZ$;
SPC( 11 -  LEN ( STR$ (DX)));DX
900  PRINT D$;P$
910  RETURN
920  PRINT  LEFT$ (PX$,6); TAB( 8); LEFT$ (Q$,18); TAB( 28);AZ$; SPC( 9 -  LEN ( S
TR$ (DX)));DX
930  RETURN
940  PRINT : PRINT : PRINT : PRINT  SPC( 16);B1$
950  PRINT : PRINT  SPC( 16);B2$
960  PRINT : PRINT  SPC( 9);"BEAM HEADINGS AND DISTANCES CUSTOMIZED FOR ";CL$
970  PRINT  SPC( 4);"LOCATED AT ";LA;"-";AM;NO$;" AND ";LO;"-";OM;WE$
980  PRINT : PRINT
990  PRINT  SPC( 4);"ALL MEASUREMENTS ARE TO THE GEOGRAPHIC CENTER OF THE COUNTRY"
1000  PRINT  SPC( 4);"WHEN NO CITY IS SPECIFIED.  HEADINGS ARE IN DEGREES EAST OF"
1010  PRINT  SPC( 4);"NORTH, AND DISTANCES ARE IN STATUTE MILES VIA THE SHORT PATH
"
1020  PRINT : PRINT : PRINT F$: PRINT
1030  PRINT D$;P$
1040  RETURN
1080  IF LT > A1 THEN AZ = 180: GOTO 606
1082  IF LT < A1 THEN AZ = 0: GOTO 606
1084  IF LT > A1 THEN DX = LT - A1: GOTO 630
1086  IF LT < A1 THEN DX = A1 - LT: GOTO 630
```

BEAMHEADING CHART (Commodore 64)

```
 5 OPEN 130,2,2,CHR$(136):REM 1200B/2STOP
 10 REM BEAM ANTENNA HEADING AND DISTANCE CHART PROGRAM (COMMODORE)
 20 REM BY KC6A AND N6NB;(C)1983 BY HAYDEN BOOK CO.,INC.
 30 CLS$=CHR$(147)
 32 PRINT CLS$:PRINT
 40 PRINT "    BEAM HEADING AND DISTANCE PROGRAM"
 50 PRINT "     FOR MAJOR CITIES, DXCC COUNTRIES"
 55 PRINT:PRINT "     BE SURE 'LATLONG' FILE IS ON DISK"
 60 PRINT:PRINT:PRINT
 62 INPUT"WHAT IS YOUR CALL ";CL$
 64 PRINT:INPUT "WANT A HARD COPY ";LP$
 66 PRINT:PRINT
 70 DIM PX$(500),Q$(500),AX(500),OX(500)
 80 RAD=57.2958:PI=3.14159
 90 F$="      PREFIX          LOCATION                BEARING          RANGE"
 100 PRINT "ENTER YOUR COORDINATES:"
 110 PRINT "   USE '-' FOR BOTH DEGREES AND MINS"
 120 PRINT "   IF SOUTH LATITUDE, EAST LONGITUDE"
 125 PRINT:PRINT "   EXAMPLE: 33,15 OR -33,-15"
 130 PRINT:INPUT "LATITUDE  ";LA,AM
 140 INPUT "LONGITUDE ";LO,OM
 150 PRINT:PRINT
 160 PRINT "ALL MEASUREMENTS ARE TO THE CENTER OF"
```

```
162 PRINT "THE COUNTRY WHEN NO CITY IS SPECIFIED."
164 PRINT "HEADINGS ARE IN DEGREES EAST OF NORTH"
166 PRINT "AND DISTANCES ARE IN STATUTE MILES"
170 PRINT "VIA SHORT PATH."
180 PRINT:PRINT
190 PRINT "NOW READING IN DATA FOR ALL COUNTRIES"
200 PRINT "          PLEASE STAND BY..."
210 PRINT:PRINT:PRINT
220 LT=(LA+AM/60)/RAD
230 LG=(LO+OM/60)/RAD
240 OPEN 8,8,8,"LATLONG,S,R"
250 J=1
260 INPUT#8,PX$(J),Q$(J),AX(J),OX(J)
270 IF PX$(J)="ZZZ" THEN PX$(J)="":Q$(J)="":GOTO 300
280 J=J+1:GOTO 260
300 CLOSE 8
302 NO$="N LAT":IF LT<0 THEN NO$="S LAT"
304 WE$="W LONG":IF LG<0 THEN WE$="E LONG"
310 IF LP$<>"Y" THEN 360
312 CMD 130:PRINT TAB(10)"CUSTOM BEAM HEADING AND DISTANCE CHART FOR ";CL$
314 PRINT:PRINT
316 PRINT TAB(10)"LOCATED AT ";LA;"-";AM;NO$;" AND ";LO;"-";OM;WE$
318 PRINT:PRINT:PRINT
320 PRINT TAB(5) "ALL MEASUREMENTS ARE TO THE CENTER OF THE COUNTRY WHEN"
322 PRINT TAB(5)"NO CITY IS SPECIFIED.  HEADINGS ARE IN DEGREES EAST OF"
324 PRINT TAB(5)"NORTH, AND DISTANCES ARE IN STATUTE MILES VIA SHORT PATH."
326 PRINT:PRINT
330 PRINT F$:PRINT#130
332 LN=15
360 N=J-1
370 FOR J=1 TO N
380 A1=AX(J)/RAD:O1=OX(J)/RAD
390 L=LG-O1:IF L=0 THEN 570
400 IF L>PI THEN L=L-2*PI
410 IF L<-PI THEN L=L+2*PI
420 AA=COS(L)*(COS(A1)/SIN(A1)):AA=ATN(AA)
430 AZ=((COS(L)/SIN(L))*COS(LT+AA))/SIN(AA):AZ=ATN(1/AZ)
440 AZ=AZ*RAD:L=L*RAD
450 IF L>0 AND AZ<0 THEN AZ=AZ+180
460 IF L<0 AND AZ>0 THEN AZ=AZ+180
470 IF L<0 AND AZ<0 THEN AZ=AZ+360
480 AZ=INT(AZ+.5)
490 DX=(SIN(LT)*SIN(A1))+(COS(LT)*COS(A1)*COS(L/RAD))
500 DX=SQR(1-(DX*DX))/DX
510 IF LG=O1 THEN 590
520 DX=ATN(DX)
530 DX=DX*RAD:IF DX<0 AND DX>-90 THEN DX=DX+180
540 DX=DX*69.0468
550 DX=INT(DX+.5)
551 A$=PX$(J):B$=LEFT$(Q$(J),20):C$=STR$(AZ):D$=STR$(DX)
552 PRINT A$;TAB(6)B$;TAB(28)C$;TAB(33)D$
553 IF LP$<>"Y" THEN 560
554 PRINT#130,"     "A$SPC(12-LEN(A$))B$SPC(24-LEN(B$))C$SPC(13-LEN(C$))D$
555 LN=LN+1
556 IF LN=>60 THEN GOSUB 900
560 NEXT J
562 IF LP$="Y" THEN GOSUB 920
564 FOR J=1TO1000:PRINT;:NEXT J
566 CLOSE 130
568 END
570 IF LT>A1 THEN AZ=180:GOTO 480
580 IF LT<A1 THEN AZ=0:GOTO 480
590 IF LT>A1 THEN DX=LT-A1:GOTO 530
600 IF LT<A1 THEN DX=A1-LT:GOTO 530
900 FOR LF=1 TO 11:PRINT#130:NEXT
905 PRINT#130,F$:PRINT#130:LN=7
910 RETURN
```

```
920 FOR FF=LN TO 65:PRINT#130:NEXT FF
925 RETURN
```

THE ON-LINE DX DISPLAY PROGRAM

What it does

The DX DISPLAY program is a beamheading chart program without the chart. It reads in the data from the LATLONG file and stores it in memory. When you type any prefix it recognizes, it will display the prefix and country (or city) and the distance and beamheading to that place from your location. It's an on-line program, one that can be used while you're operating. When you hear a particular country, enter the prefix to see a display of the particulars.

Setup instructions

This program will run on most computers with few modifications if any. About the only thing it does that varies from computer to computer is calling up the LATLONG data file. And, of course, the program will work with or without disk drives, since the data file can be converted into DATA statements within the program (as explained earlier in this chapter).

However, if you change the length of the LATLONG file, the prefix searching routine in lines 672–678 will have to be changed or deleted. The routine looks at the prefix that has been entered and jumps to various places in the data to begin searching. That is done to increase the response speed; the program will work correctly (but more slowly) if lines 672–678 are deleted.

If you significantly change the length of the LATLONG file, it is advisable to change lines 672–678. Line 672 tests to see if the prefix begins with a numeral. If it does, the search begins at line 350 in the list of prefixes, countries and cities. If the prefixes beginning with numerals start before 350 in your list, change the value of L accordingly in line 672.

Use the same procedure to modify lines 674, 676, and 678. Line 674 is the starting point if the call begins with W, X, Y, or Z. If your first "W" prefix comes before the 250th entry in the data file, change the value of L accordingly. Line 676 begins the search at the 170th line if a prefix begins with "U" or "V." If your first "U" is before the 170th entry, change the value of L in line 676. And 678 begins the search at the 85th entry for calls beginning with J through T. If your first "J" prefix is before the 85th line of the data file, change line 678.

Experienced programmers will note that there are a number of other ways the 400-plus items in the prefix array could be searched, including some that are faster than this method. There could be a separate index that tells where to begin the search, for example. However, most computers will find any prefix within 1 to 2 seconds with this routine, so additional complications were judged unnecessary.

Operating notes

To use this program, enter your latitude and longitude, again using a minus sign for both degrees AND minutes to indicate east longitude or south latitude.

Once you've entered your coordinates, the LATLONG file will be loaded into large arrays in memory, a process that takes a minute or two on most computers. Then you'll see a menu displayed, giving you several options.

The first option is to enter just a prefix. The program will search for that prefix and display the country (or city) and compute the distance and direction. If there is more than one entry in the data file with that prefix, all will be displayed.

You can enter just a single letter such as "V," and you'll get a display of all entries that have "V" as the first letter of their prefix. If you enter "W," be prepared for a lengthy display.

One thing to remember is that the program will only display prefixes that match those in the data file. Many countries have more than one prefix, but the data file uses only the primary prefix for each one in most cases. For example, if you type "JH" for Japan, it won't work. The program doesn't know all of the additional prefixes for various countries. In general, use just the first two letters without any numeral, unless you're sure the particular numeral is unique. But in some cases, not only the number but also the first letter of the suffix is specific to a country and will match the data file's prefix list. VK9X, for instance, is the "prefix" for Christmas Island.

For large countries, there is typically more than one entry in the data file, each with its own prefix. Thus, if you type "VE1" or "UA1" you'll get a different place within Canada or the Soviet Union than you will get if you type "VE3" or "UA3."

If you type a prefix that doesn't exist in the data file, all that will happen is a 5-to-15-second delay while your prefix is compared to all possible prefixes.

If you're not sure what prefix to use for a particular country, simply type "?" and you'll get a display of all acceptable prefixes.

Modifications for CP/M

Modify the TRS-80 and CP/M listing as follows for CP/M systems running BASIC-80:

1. Delete the CLEAR command in line 70;
2. Replace the clear-screen command in lines 32 and 610 with the proper command for your system.

DX DISPLAY (TRS-80 AND CP/M)

```
10 REM ANTENNA HEADING AND DISTANCE DISPLAY (TRS-80 VERSION)
20 REM BY KC6A AND N6NB
30 REM (C) 1983 BY HAYDEN BOOK CO.,INC.
32 CLS:REM CLEAR SCREEN
40 B1$="ON-LINE BEAM HEADING AND DISTANCE DISPLAY"
50 B2$="   FOR DXCC COUNTRIES AND MAJOR CITIES"
60 PRINT TAB(12) B1$:PRINT TAB(12) B2$:PRINT:PRINT
70 CLEAR 15000
80 PRINT "BE SURE THE DATA FILE (LATLONG) IS ON DISK"
90 PRINT:PRINT:PRINT
```

```
100 DIM PX$(500):DIM Q$(500):DIM AX(500):DIM OX(500)
110 PI=3.14159:RAD=57.2958
120 REM TO CUSTOMIZE, DELETE LINES 140-210 AND SET LA,AM,LO,OM
130 REM EQUAL TO YOUR LATITUDE AND LONGITUDE IN DEGREES,MINS
140 PRINT "ENTER YOUR COORDINATES:"
142 PRINT "   USING '-' FOR BOTH DEGREES AND MINUTES"
144 PRINT "   IF SOUTH LATITUDE OR EAST LONGITUDE"
150 PRINT
160 INPUT "LATITUDE:  (E.G. 33,30)";LA,AM
170 INPUT "LONGITUDE:  (E.G.116,30)";LO,OM
180 IF LA<90 AND LA>-90 AND LO<180 AND LO>-180 GOTO 220
190 PRINT "NOTE:   LATITUDE MUST BE > -90 AND < +90."
200 PRINT "        LONGITUDE MUST BE > -180 AND < +180."
210 GOTO 160
220 PRINT:PRINT
230 PRINT "ALL MEASUREMENTS ARE TO THE CENTER OF THE COUNTRY WHEN"
240 PRINT "NO CITY IS SPECIFIED.  HEADINGS ARE IN DEGREES EAST OF"
250 PRINT "NORTH, AND DISTANCES ARE IN STATUTE MILES VIA SHORT PATH."
260 PRINT:PRINT
270 PRINT TAB(8) "NOW READING IN DATA FOR ALL COUNTRIES"
280 PRINT TAB(8) "         PLEASE STAND BY..."
290 PRINT:PRINT:PRINT
300 LT=(LA+AM/60)/RAD
310 LG=(LO+OM/60)/RAD
320 OPEN "I",1,"LATLONG"
330 J=1
350 INPUT #1, PX$(J),Q$(J),AX(J),OX(J)
352 IF PX$(J)="ZZZ" THEN PX$(J)="":Q$(J)="":GOTO 370
360 J=J+1:GOTO 350
370 CLOSE:GOTO 610
380 A1=AX(J)/RAD:O1=OX(J)/RAD
390 L=LG-O1:IF L=0 THEN 570
400 IF L>PI THEN L=L-2*PI
410 IF L<-PI THEN L=L+2*PI
420 AA=COS(L)*(COS(A1)/SIN(A1)):AA=ATN(AA)
430 AZ=((COS(L)/SIN(L))*COS(LT+AA))/SIN(AA):AZ=ATN(1/AZ)
440 AZ=AZ*RAD:L=L*RAD
450 IF L>0 AND AZ<0 THEN AZ=AZ+180
460 IF L<0 AND AZ>0 THEN AZ=AZ+180
470 IF L<0 AND AZ<0 THEN AZ=AZ+360
480 AZ=INT(AZ+.5)
490 DX=(SIN(LT)*SIN(A1))+(COS(LT)*COS(A1)*COS(L/RAD))
500 DX=SQR(1-(DX*DX))/DX
510 IF LG=O1 THEN 590
520 DX=ATN(DX)
530 DX=DX*RAD:IF DX<0 AND DX>-90 THEN DX=DX+180
540 DX=DX*69.0468
550 DX=INT(DX+.5)
560 RETURN
570 IF LT>A1 THEN AZ=180:GOTO 480
580 IF LT<A1 THEN AZ=0:GOTO 480
590 IF LT>A1 THEN DX=LT-A1:GOTO 530
600 IF LT<A1 THEN DX=A1-LT:GOTO 530
610 CLS
618 PRINT TAB(18)STRING$(22,42)
620 PRINT TAB(18)"OPERATING INSTRUCTIONS"
621 PRINT TAB(18)STRING$(22,42):PRINT
622 PRINT TAB(9)"ENTER DESIRED PREFIX, OR:"
630 PRINT TAB(12)"-TO DISPLAY ALL ACCEPTABLE PREFIXES TYPE '?'""
634 PRINT TAB(12)"-TO DISPLAY PREFIXES STARTING WITH"
636 PRINT TAB(15)"A CERTAIN LETTER, TYPE THE LETTER"
640 PRINT TAB(12)"-TO EXIT FROM PROGRAM, TYPE 'EX'"
642 INPUT "COMMAND";P1$
650 PRINT:PRINT
660 IF P1$="EX" THEN 800
662 LN=0
670 IF P1$="?" THEN 810
672 IF ASC(LEFT$(P1$,1))<58 THEN L=350:GOTO 680
674 IF ASC(LEFT$(P1$,1))=>87 THEN L=250:GOTO 680
676 IF ASC(LEFT$(P1$,1))=>85 THEN L=170:GOTO 680
```

```
678 IF ASC(LEFT$(P1$,1))=>74 THEN L=85
680 FOR J=L TO 500
685 IF P1$<>LEFT$(PX$(J),LEN(P1$)) AND X=1 THEN 780
690 IF P1$<>LEFT$(PX$(J),LEN(P1$)) THEN 740
700 PRINT "FOR ";PX$(J);" AT ";Q$(J)
710 GOSUB 380
720 PRINT "HEADING: ";AZ;"DEGREES;    DISTANCE: ";DX;"STAT.MILES"
730 PRINT:X=1:LN=LN+3
735 IF LN=>12 THEN GOSUB 880
740 NEXT J
760 PRINT "PREFIX NOT FOUND IN COUNTRIES LIST"
770 PRINT "...TRY USING PRIMARY PREFIX WITHOUT A CALL AREA NUMERAL"
780 PRINT:PRINT:X=0:L=1
782 IF LN>3 THEN INPUT "<ENTER> TO CONTINUE";GO$
790 GOTO 618
800 END
810 LN=0
820 FOR J=1 TO 500
830 IF Q$(J)="" THEN GOTO 610
840 LN=LN+1
850 IF LN=12 THEN GOSUB 880
860 PRINT PX$(J);TAB(12-LEN(PX$(J)));Q$(J)
870 NEXT J
880 LN=1:INPUT "<ENTER> TO CONTINUE";GO$
882 RETURN
```

DX DISPLAY (Apple)

```
 10  REM  ANTENNA HEADING AND DISTANCE DISPLAY (APPLE)
 20  REM  BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
 30  FOR J = 1 TO 24: PRINT : NEXT J
 40 B1$ = "RAPID BEAM HEADING AND DISTANCE DISPLAY"
 50 B2$ = "  FOR DXCC COUNTRIES AND MAJOR CITIES"
 60  PRINT B1$: PRINT B2$: PRINT : PRINT
 70  FOR J = 1 TO 10: PRINT : NEXT J
 80  PRINT "PLEASE BE SURE THE DATA FILE"
 82  PRINT "(LATLONG DATA) IS ON LOGGED DISK"
 90  PRINT : PRINT : PRINT
100  DIM PX$(500): DIM Q$(500): DIM AX(500): DIM OX(500)
110 PI = 3.14159:RAD = 57.2958:D$ =  CHR$ (4)
120  REM  TO CUSTOMIZE THIS, DELETE LINES 140-210 AND SET VARIABLES LA,AM,LO,OM
130  REM  EQUAL TO YOUR LATITUDE AND LONGITUDE (IN DEGREES, MINUTES)
140  PRINT "ENTER YOUR COORDINATES, USING '-'"
142  PRINT "FOR SOUTH LATITUDE, EAST LONGITUDE"
150  PRINT
160  INPUT "LATITUDE:  (E.G. 33,30)";LA,AM
170  INPUT "LONGITUDE:  (E.G.116,30)";LO,OM
180  IF LA < 90 AND LA > -90 AND LO < 180 AND LO > - 180 GOTO 220
190  PRINT "PLEASE NOTE:   LATITUDE MUST BE MORE THAN -90 AND LESS THAN +90."
200  PRINT "               LONGITUDE MUST BE MORE THAN -180 AND LESS THAN +180."
210  GOTO 160
220  PRINT : PRINT
230  PRINT "ALL MEASUREMENTS ARE TO THE CENTER OF"
232  PRINT "THE COUNTRY WHEN NO CITY IS LISTED."
240  PRINT "HEADINGS ARE IN DEGREES EAST OF NORTH"
250  PRINT "AND DISTANCES ARE VIA SHORT PATH."
260  PRINT : PRINT
270  PRINT "NOW READING IN DATA FOR ALL COUNTRIES"
280  PRINT "        PLEASE STAND BY..."
290  PRINT : PRINT : PRINT
300 LT = (LA + AM / 60) / RAD
310 LG = (LO + OM / 60) / RAD
320  PRINT D$;"OPEN LATLONG DATA"
330  FOR J = 1 TO 500
```

```
340  PRINT D$;"READ LATLONG DATA"
350  INPUT PX$(J),Q$(J),AX(J),OX(J)
355  IF PX$(J) = "ZZZ" THEN PX$(J) = "":Q$(J) = "": GOTO 370
360  NEXT J
370  PRINT D$;"CLOSE": GOTO 620
380 A1 = AX(J) / RAD:O1 = OX(J) / RAD
390 L = LG - O1: IF L = 0 THEN 570
400  IF L > PI THEN L = L - 2 * PI
410  IF L <  - PI THEN L = L + 2 * PI
420 AA =  COS (L) * ( COS (A1) /  SIN (A1)):AA =  ATN (AA)
430 AZ = (( COS (L) /  SIN (L)) *  COS (LT + AA)) /  SIN (AA):AZ =  ATN (1 / AZ)
440 AZ = AZ * RAD:L = L * RAD
450  IF L > 0 AND AZ < 0 THEN AZ = AZ + 180
460  IF L < 0 AND AZ > 0 THEN AZ = AZ + 180
470  IF L < 0 AND AZ < 0 THEN AZ = AZ + 360
480 AZ =  INT (AZ + .5)
490 DX = ( SIN (LT) *  SIN (A1)) + ( COS (LT) *  COS (A1) *  COS (L / RAD))
500 DX =  SQR (1 - (DX * DX)) / DX
510  IF LG = O1 THEN 590
520 DX =  ATN (DX)
530 DX = DX * RAD: IF DX < 0 AND DX >  - 90 THEN DX = DX + 180
540 DX = DX * 69.0468
550 DX =  INT (DX + .5)
560  RETURN
570  IF LT > A1 THEN AZ = 180: GOTO 480
580  IF LT < A1 THEN AZ = 0: GOTO 480
590  IF LT > A1 THEN DX = LT - A1: GOTO 530
600  IF LT < A1 THEN DX = A1 - LT: GOTO 530
610  PRINT : PRINT : PRINT
620  PRINT "         OPERATING INSTRUCTIONS": PRINT
622  PRINT "   -ENTER DESIRED PREFIX, OR:"
630  PRINT "   -TO DISPLAY ALL ACCEPTABLE PREFIXES"
632  PRINT "      TYPE '?'"
634  PRINT "   -TO DISPLAY PREFIXES STARTING"
636  PRINT "      WITH A LETTER, TYPE THE LETTER"
640  INPUT "   -TO EXIT FROM PROGRAM, TYPE 'E' ";P1$
650  PRINT : PRINT
660  IF P1$ = "E" THEN 800
665 LN = 0
670  IF P1$ = "?" THEN 810
672  IF  ASC ( LEFT$ (P1$,1)) < 58 THEN L = 350: GOTO 680
674  IF  ASC ( LEFT$ (P1$,1)) = 87 THEN L = 250: GOTO 680
676  IF  ASC ( LEFT$ (P1$,1)) =  > 85 THEN L = 170: GOTO 680
678  IF  ASC ( LEFT$ (P1$,1)) =  > 74 THEN L = 85
680  FOR J = L TO 500
685  IF P1$ <  >  LEFT$ (PX$(J), LEN (P1$)) AND X = 1 THEN 780
690  IF P1$ <  >  LEFT$ (PX$(J), LEN (P1$)) THEN 740
700  PRINT "FOR ";PX$(J);" AT ";Q$(J)
710  GOSUB 380
720  PRINT  SPC( 5)"BEAM HEADING: ";AZ;" DEGREES"
725  PRINT  SPC( 5)"DISTANCE: ";DX;" STAT. MILES"
730  PRINT :X = 1:LN = LN + 4
735  IF LN = 20 THEN  GOSUB 880
740  NEXT J
750  IF X = 1 THEN 780
760  PRINT "PREFIX NOT FOUND IN COUNTRIES LIST"
770  PRINT "...TRY USING PRIMARY PREFIX WITHOUT A CALL AREA NUMERAL"
780  PRINT : PRINT : PRINT :X = 0:L = 1
790  GOTO 620
800  END
810  FOR J = 1 TO 500
830  IF Q$(J) = "" THEN 610
840 LN = LN + 1
850  IF LN = 24 THEN  GOSUB 880
860  PRINT PX$(J); SPC( 12 -  LEN (PX$(J)));Q$(J)
870  NEXT J
880 LN = 1: INPUT "ENTER <CR> TO CONTINUE";GO$: RETURN
```

DX DISPLAY (Commodore 64)

```
10 REM ON-LINE HEADING/DX CHECKER (C64)
20 REM BY KC6A AND N6NB;(C)1983 BY HAYDEN BOOK CO.,INC.
30 CLS$=CHR$(147)
32 PRINT CLS$:PRINT
40 PRINT "      ON-LINE HEADING AND DX DISPLAY"
50 PRINT "     FOR MAJOR CITIES, DXCC COUNTRIES"
55 PRINT:PRINT "     BE SURE 'LATLONG' FILE IS ON DISK"
60 PRINT:PRINT:PRINT
70 DIM PX$(500),Q$(500),AX(500),OX(500)
80 RAD=57.2958:PI=3.14159
100 PRINT "ENTER YOUR COORDINATES:"
110 PRINT "   USE '-' FOR BOTH DEGREES AND MINS"
120 PRINT "   IF SOUTH LATITUDE, EAST LONGITUDE"
130 PRINT:INPUT "LATITUDE  ";LA,AM
140 INPUT "LONGITUDE ";LO,OM
150 PRINT:PRINT
160 PRINT "ALL MEASUREMENTS ARE TO THE CENTER OF"
162 PRINT "THE COUNTRY WHEN NO CITY IS SPECIFIED."
164 PRINT "HEADINGS ARE IN DEGREES EAST OF NORTH"
166 PRINT "AND DISTANCES ARE IN STATUTE MILES"
170 PRINT "VIA SHORT PATH."
180 PRINT:PRINT
190 PRINT "NOW READING IN DATA FOR ALL COUNTRIES"
200 PRINT "          PLEASE STAND BY..."
210 PRINT:PRINT:PRINT
220 LT=(LA+AM/60)/RAD
230 LG=(LO+OM/60)/RAD
240 OPEN 8,8,8,"LATLONG,S,R"
250 J=1
260 INPUT#8,PX$(J),Q$(J),AX(J),OX(J)
270 IF PX$(J)="ZZZ" THEN PX$(J)="":Q$(J)="":GOTO 300
280 J=J+1:GOTO 260
300 CLOSE 8
310 GOTO 610
380 A1=AX(J)/RAD:O1=OX(J)/RAD
390 L=LG-O1:IF L=0 THEN 570
400 IF L>PI THEN L=L-2*PI
410 IF L<-PI THEN L=L+2*PI
420 AA=COS(L)*(COS(A1)/SIN(A1)):AA=ATN(AA)
430 AZ=((COS(L)/SIN(L))*COS(LT+AA))/SIN(AA):AZ=ATN(1/AZ)
440 AZ=AZ*RAD:L=L*RAD
450 IF L>0 AND AZ<0 THEN AZ=AZ+180
460 IF L<0 AND AZ>0 THEN AZ=AZ+180
470 IF L<0 AND AZ<0 THEN AZ=AZ+360
480 AZ=INT(AZ+.5)
490 DX=(SIN(LT)*SIN(A1))+(COS(LT)*COS(A1)*COS(L/RAD))
500 DX=SQR(1-(DX*DX))/DX
510 IF LG=O1 THEN 590
520 DX=ATN(DX)
530 DX=DX*RAD:IF DX<0 AND DX>-90 THEN DX=DX+180
540 DX=DX*69.0468
550 DX=INT(DX+.5)
560 RETURN
570 IF LT>A1 THEN AZ=180:GOTO 480
580 IF LT<A1 THEN AZ=0:GOTO 480
590 IF LT>A1 THEN DX=LT-A1:GOTO 530
600 IF LT<A1 THEN DX=A1-LT:GOTO 530
610 PRINT CLS$
620 PRINT:PRINT TAB(8)"OPERATING INSTRUCTIONS"
621 PRINT
622 PRINT TAB(5)"ENTER DESIRED PREFIX OR-"
623 PRINT TAB(8)"TO DISPLAY ALL ACCEPTABLE"
624 PRINT TAB(8)"   PREFIXES TYPE '?'"
625 PRINT TAB(8)"-TO DISPLAY PREFIXES STARTING"
626 PRINT TAB(8)"   WITH A LETTER, TYPE IT"
627 PRINT TAB(8)"-TO EXIT, TYPE 'E'"
628 PRINT:INPUT "COMMAND";P1$
640 PRINT:PRINT
```

```
660 IF P1$="E" THEN 800
662 LN=0
670 IF P1$="?" THEN 810
672 IF ASC(LEFT$(P1$,1))<58 THEN L=350:GOTO 680
674 IF ASC(LEFT$(P1$,1))=>87 THEN L=250:GOTO 680
676 IF ASC(LEFT$(P1$,1))=>85 THEN L=170:GOTO 680
678 IF ASC(LEFT$(P1$,1))=>74 THEN L=85
680 FOR J=L TO 500
685 IF P1$<>LEFT$(PX$(J),LEN(P1$)) AND X=1 THEN 780
690 IF P1$<>LEFT$(PX$(J),LEN(P1$)) THEN 740
700 PRINT "FOR ";PX$(J);" AT ";Q$(J)
710 GOSUB 380
720 PRINT "AZIMUTH: ";AZ;"  DX:  ";DX;" STAT.MILES"
730 PRINT:X=1:LN=LN+3
735 IF LN=>12 THEN GOSUB 880
740 NEXT J
760 PRINT "PREFIX NOT FOUND IN COUNTRIES LIST"
770 PRINT "...TRY USING PRIMARY PREFIX WITHOUT"
772 PRINT "   A CALL AREA NUMERAL"
780 PRINT:PRINT:X=0:L=1
782 IF LN=>15 THEN INPUT "<CR> TO CONTINUE";GO$
790 GOTO 620
800 END
810 LN=0
820 FOR J=1 TO 500
830 IF Q$(J)="" THEN 782
840 LN=LN+1
850 IF LN=>20 THEN GOSUB 880
860 PRINT PX$(J);TAB(8)Q$(J)
870 NEXT J
880 LN=1:INPUT "<CR> TO CONTINUE";GO$
882 RETURN
```

THE SUNRISE CHART PROGRAM

What it does

The SUNRISE CHART program, as indicated earlier, reads in the LATLONG data file and computes the sunrise and sunset times for all countries and cities on the list, displaying (or printing) the results for any one day.

Setup instructions

An important point to remember about this and the other sunrise-based programs: the formula used to compute sunrise and sunset times is based on *actual sunrise times,* not *apparent sunrise/sunset times,* which differ slightly due to atmospheric refraction. As the sun nears the horizon, its light must pass through many miles of the earth's atmosphere. In a phenomenon not unlike the one that sometimes produces dramatic tropospheric DX on the VHF bands, the sun appears to be above the horizon for a few minutes when it actually isn't. For that reason, the apparent sunrise is always a little earlier—and the sunset a little later—than these sunrise programs indicate.

If you prefer apparent sunrise times to actual ones, you can approximate those times by adding three minutes to S2 in line 650 on the TRS-80 and CP/M listing or line 700 of the Apple version, and by *subtracting* a like amount from R2 in line 600 of the TRS-80 listing or 660 of the Apple version. *To derive apparent rather than actual sunrise/sunset*

times, a similar change should be made in each of the sunrise-based programs.

Operating notes

To run the program, you just enter the month and day and indicate whether you want a hard-copy printout—that's all.

Modifications for CP/M

The TRS-80 and CP/M listing should be modified as follows for CP/M systems using BASIC-80:

1. Delete the CLEAR statement in line 8.

2. Replace the clear-screen statement in line 80 with the proper command for your system.

3. The print-using format variables in lines 280 and 290 must have a back-slash ("\") in place of each percent sign ("%").

SUNRISE CHART (TRS-80 and CP/M)

```
5 REM SUNRISE CHART PROGRAM FOR ONE DAY, MANY LOCATIONS (TRS-80)
7 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
8 CLEAR 250
10 B1$= "SUNRISE/SUNSET CALCULATION PROGRAM"
20 B2$= "FOR DXCC COUNTRIES PLUS MAJOR CITIES"
30 B3$= "ALL TIMES ARE IN GMT; SOME MAY BE NEXT (OR PREVIOUS) GMT DAY"
40 REM FOR THIS PROGRAM, THE DATA FILE (LATLONG) MUST BE ON DISK
60 REM TIMES GIVEN ARE ACTUAL SUNRISE/SUNSET.  APPARENT TIMES
70 REM DIFFER BY 5-7 MINS DUE TO ATMOSPHERIC REFRACTION.
80 CLS:REM CLEAR SCREEN
90 F$="     PREFIX        LOCATION          SUNRISE   SUNSET  HRS SUN"
100 PRINT TAB(9)B1$
110 PRINT TAB(8)B2$:PRINT:PRINT
120 INPUT "ENTER MONTH AND DAY FOR THIS PRINTOUT (E.G. 7,4)";M,DA
130 PRINT:INPUT "DO YOU WANT A HARD COPY";LP$
140 PRINT:PRINT
150 IF LP$<>"Y" THEN 200
160 FOR J=1 TO 5:LPRINT:NEXT J
170 LPRINT TAB(17)B1$:LPRINT TAB(16)B2$:LPRINT:LPRINT
180 LPRINT TAB(5)"THIS PRINTOUT IS FOR THE FOLLOWING DATE:";M;"/";DA:LPRINT
190 LPRINT:LPRINT TAB(5)B3$:LPRINT:LPRINT F$:LPRINT
200 RAD=57.2958:PI=3.14159/26
210 DIM N(12)
220 FOR I=1 TO 12
230 READ N(I)
240 NEXT I
250 DATA 0,31,59,90,120,151,181,212,243,273,304,334
260 REM EXTRA DAY IN LEAP YEARS IGNORED
270 PRINT B3$:PRINT
280 F1$="     %    %  %                    %  ## ##     ## ##   ## ##"
290 F2$="     %    %  %                  %  %                      %"
300 A$= "NO SUNSET TODAY HERE"
310 B$= "NO SUNRISE TODAY HERE"
320 PRINT F$:PRINT:LN=18
330 OPEN "I",1,"LATLONG"
340 X=PI*(N(M)+DA)/7
350 D=.456-22.915*COS(X)-.43*COS(2*X)-.156*COS(3*X)+3.83*SIN(X)+.06*SIN(2*X)-.08
    2*SIN(3*X)
360 E=.008+.51*COS(X)-3.197*COS(2*X)-.106*COS(3*X)-.15*COS(4*X)-7.317*SIN(X)-9.4
    71*SIN(2*X)-.391*SIN(3*X)-.242*SIN(4*X)
370 REM D IS DECLINATION OF SUN IN DEGREES; E IS EQUATION OF TIME IN MINUTES
380 DE=D/RAD
390 IF EOF(1) THEN 750
400 LN=LN+1
410 IF LN=60 AND LP$="Y" THEN GOSUB 780
420 INPUT #1,PX$,Q$,LA,LO
422 IF PX$="ZZZ" THEN 750
```

```
430 LA=LA/RAD
440 Y=TAN(DE)*TAN(LA)
450 IF Y=>1 THEN 690
460 IF Y<=-1 THEN 720
470 AR=-ATN(Y/SQR(-Y*Y+1))+1.5708
480 AC=AR*RAD
490 RI=LO/15+AC/15:RI=RI-E/60
500 ST=LO/15-AC/15:ST=ST-E/60
510 LT=ST-RI
520 IF LT<0 THEN LT=24+LT
530 IF (LT-INT(LT))>.99 THEN LT=(INT((LT*100)+.6))/100
540 LH=INT(LT)
550 LM=(LT-INT(LT))*60
560 IF RI>24 THEN RI=RI-24
570 IF RI<0 THEN RI=RI+24
580 IF (RI-INT(RI))>.99 THEN RI=(INT((RI*100)+.6))/100
590 R1=INT(RI)
600 R2=(RI-INT(RI))*60
610 IF ST>24 THEN ST=ST-24
620 IF ST<0 THEN ST=ST+24
630 IF (ST-INT(ST))>.99 THEN ST=(INT((ST*100)+.6))/100
640 S1=INT(ST)
650 S2=(ST-INT(ST))*60
660 PRINT USING F1$;PX$;Q$;R1;R2;S1;S2;LH;LM
670 IF LP$="Y" THEN LPRINT USING F1$;PX$;Q$;R1;R2;S1;S2;LH;LM
680 GOTO 390
690 PRINT USING F2$;PX$;Q$;A$
700 IF LP$="Y" THEN LPRINT USING F2$;PX$;Q$;A$
710 GOTO 390
720 PRINT USING F2$;PX$;Q$;B$
730 IF LP$="Y" THEN LPRINT USING F2$;PX$;Q$;B$
740 GOTO 390
750 CLOSE
760 FOR J=LN TO 65:LPRINT:NEXT J
770 END
780 FOR J=1 TO 11:LPRINT:NEXT J
790 LPRINT F$:LPRINT:LN=7
800 RETURN
```

SUNRISE CHART (Apple)

```
10  REM  SUNRISE CHART PROGRAM FOR ONE DAY, MANY LOCATIONS (APPLE)
20  REM  PRINTS DATA FOR MAJOR CITIES/DXCC COUNTRIES ON ANY DAY
30  REM  BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
32  REM   'LATLONG DATA' FILE MUST BE ON DISK.
34  REM  TIMES GIVEN ARE ACTUAL SUNRISE/SUNSET; APPARENT TIMES
36  REM  DIFFER BY 5-7 MINS DUE TO ATMOSPHERIC REFRACTION
50  HOME
60 B1$ = " SUNRISE/SUNSET CALCULATION PROGRAM"
70 B2$ = "FOR DXCC COUNTRIES PLUS MAJOR CITIES"
80 B3$ = "ALL TIMES ARE IN GMT; SOME MAY BE NEXT  (OR PREVIOUS)  GMT DAY"
90 F$ = "    PREFIX        LOCATION                SUNRISE     SUNSET    HRS SUNLI
GHT"
100  PRINT "  "B1$: PRINT : PRINT "  "B2$: PRINT : PRINT
120  FOR J = 1 TO 10: PRINT : NEXT J
130  PRINT "ENTER MONTH AND DAY FOR THIS PRINTOUT"
140  INPUT "E.G. 7,4) ";M,DA
150  PRINT
160  INPUT "DO YOU WANT A HARD COPY PRINTOUT? (Y/N) ";LP$
190  PRINT : PRINT
200 D$ =  CHR$ (4):I$ =  CHR$ (9):P$ = "PR#0":P1$ = "PR#1":X$ = "80N"
220  IF LP$ = "Y" THEN  PRINT D$;P1$: PRINT I$;X$: GOSUB 950
230 RA = 57.2958:P = 3.14159 / 26
240  DIM N(12)
250  FOR I = 1 TO 12
260  READ N(I)
```

```
270  NEXT I
280  DATA   0,31,59,90,120,151,181,212,243,273,304,334
290  REM   EXTRA DAY IN LEAP YEARS IGNORED
310  PRINT B3$: PRINT
320  IF LP$ < > "Y" THEN  PRINT "TYPE CTRL-S TO STOP AND RESTART PROGRAM": PRINT
330 A$ = "NO SUNSET TODAY"
340 B$ = "NO SUNRISE TODAY"
360  PRINT "PREFIX    LOCATION   RISE  SET  HRS SUN"
370  PRINT
380  PRINT D$;"OPEN LATLONG DATA"
390 X = P * (N(M) + DA) / 7
400 D = .456 - 22.915 * COS (X) - .43 * COS (2 * X) - .156 * COS (3 * X) + 3.83
 * SIN (X) + .06 * SIN (2 * X) - .082 * SIN (3 * X)
405 E = .008 + .51 * COS (X) - 3.197 * COS (2 * X) - .106 * COS (3 * X) - .15 *
  COS (4 * X) - 7.317 * SIN (X) - 9.471 * SIN (2 * X) - .391 * SIN (3 * X) - .2
42 * SIN (4 * X)
410  REM  D IS DECLINATION OF SUN IN DEGREES; E IS EQUATION OF TIME IN MINUTES
420 DE = D / RA
430 LN = 16
440  PRINT D$;"READ LATLONG DATA"
450 LN = LN + 1
460  IF LN = 60 AND LP$ = "Y" THEN 900
470  INPUT PX$,Q$,LA,LO
480  IF PX$ = "ZZZ" THEN  GOTO 840
490 LA = LA / RA
500 Y =  TAN (DE) *  TAN (LA)
510  IF Y => 1 THEN 810
520  IF Y <=  - 1 THEN 810
530 AR =  -  ATN (Y /  SQR ( - Y * Y + 1)) + 1.5708
540 AC = AR * RA
550 RI = LO / 15 + AC / 15:RI = RI - E / 60
560 ST = LO / 15 - AC / 15:ST = ST - E / 60
570 LT = ST - RI
580  IF LT < 0 THEN LT = 24 + LT
590  IF (LT -  INT (LT)) > .99 THEN LT = ( INT ((LT * 100) + .6)) / 100
600 LH =  INT (LT)
610 LM = (LT -  INT (LT)) * 60
620  IF RI > 24 THEN RI = RI - 24
630  IF RI < 0 THEN RI = RI + 24
640  IF (RI -  INT (RI)) > .99 THEN RI = ( INT ((RI * 100) + .6)) / 100
650 R1 =  INT (RI)
660 R2 = (RI -  INT (RI)) * 60
670  IF ST > 24 THEN ST = ST - 24
680  IF ST < 0 THEN ST = ST + 24
690  IF (ST -  INT (ST)) > .99 THEN ST = ( INT ((ST * 100) + .6)) / 100
700 S1 =  INT (ST):S2 = (ST -  INT (ST)) * 60
710 R2 =  INT (R2):S2 =  INT (S2):LM =  INT (LM):D =  INT (D * 100 + .5) / 100
720 R1$ =  STR$ (R1):R2$ =  STR$ (R2):S1$ =  STR$ (S1):S2$ =  STR$ (S2):LH$ =  STR
$ (LH):LM$ =  STR$ (LM)
730  IF R1 < 10 THEN R1$ = "0" +  STR$ (R1)
740  IF R2 < 10 THEN R2$ = "0" +  STR$ (R2)
750  IF S1 < 10 THEN S1$ = "0" +  STR$ (S1)
760  IF S2 < 10 THEN S2$ = "0" +  STR$ (S2)
770  IF LH < 10 THEN LH$ = "0" +  STR$ (LH)
780  IF LM < 10 THEN LM$ = "0" +  STR$ (LM)
790 RX$ = R1$ + ":" + R2$:SX$ = S1$ + ":" + S2$:LX$ = LH$ + ":" + LM$
810  IF LP$ = "Y" THEN  PRINT D$;P1$: GOSUB 1020
820  GOSUB 1070
830  GOTO 440
840  PRINT D$;"CLOSE"
850  IF LP$ <  > "Y" GOTO 890
860  PRINT D$;P1$
870  FOR J = LN TO 65: PRINT : NEXT J
880  PRINT D$;P$
890  END
900  PRINT D$;P1$
910  FOR J = 1 TO 11: PRINT : NEXT J
920  PRINT F$: PRINT :LN = 6
930  PRINT D$;P$
940  GOTO 440
```

```
950  PRINT : PRINT : PRINT : PRINT  TAB( 20);B1$
960  PRINT : PRINT  TAB( 20);B2$: PRINT : PRINT
970  PRINT  TAB( 15);"THIS PRINTOUT IS FOR THE FOLLOWING DATE: ";M;"/";DA
980  PRINT : PRINT : PRINT  TAB( 10);B3$
990  PRINT : PRINT : PRINT F$: PRINT
1000  PRINT D$;P$
1010  RETURN
1020  IF Y > 1 THEN  PRINT I$;X$; SPC( 4)PX$; SPC( 9 -  LEN (PX$));Q$; SPC( 30 -
LEN (Q$));A$: GOTO 1050
1030  IF Y <  - 1 THEN  PRINT I$;X$; SPC( 4)PX$; SPC( 9 -  LEN (PX$));Q$; SPC( 30
-  LEN (Q$));B$: GOTO 1050
1040  PRINT I$;X$; SPC( 4);PX$; SPC( 9 -  LEN (PX$));Q$; SPC( 30 -  LEN (Q$));RX$;
 SPC( 6);SX$; SPC( 8);LX$
1050  PRINT D$;P$
1060  RETURN
1070  IF Y > 1 THEN  PRINT  LEFT$ (PX$,5); TAB( 6); LEFT$ (Q$,14); TAB( 22);A$: GO
TO 1100
1080  IF Y <  - 1 THEN  PRINT  LEFT$ (PX$,5); TAB( 6); LEFT$ (Q$,14); TAB( 22);B$;
 GOTO 1100
1090  PRINT  LEFT$ (PX$,5); TAB( 6); LEFT$ (Q$,14); TAB( 21);RX$; TAB( 28);SX$; TA
B( 35);LX$
1100  RETURN
```

SUNRISE CHART (Commodore 64)

```
10 REM WORLDWIDE SUNRISE CHART PROGRAM (C64)
12 REM PRINTS A CHART OF SUNRISE/SUNSET TIMES FOR ANY ONE DAY
14 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
20 OPEN15,8,15
22 OPEN130,2,2,CHR$(136)+CHR$(1):REM 1200-BAUD RS232 PRINTER CHANNEL
30 B1$= "SUNRISE/SUNSET CALCULATION PROGRAM"
32 B2$= "FOR DXCC COUNTRIES PLUS MAJOR CITIES"
34 B3$= "     ALL TIMES ARE IN GMT; SOME MAY BE NEXT (OR PREVIOUS) GMT DAY"
40 REM LATLONG DATA FILE MUST BE ON LOGGED DISK DRIVE.
60 REM TIMES GIVEN ARE FOR ACTUAL SUNRISE/SUNSET; APPARENT TIMES DIFFER
70 REM BY 5-7 MINS. DUE TO ATMOSPHERIC REFRACTION.
80 PRINT CHR$(147):PRINT:REM CLEAR SCREEN
90 F$="     PREFIX          LOCATION                SUNRISE"
92 F$=F$+"    SUNSET   HRS SUNLIGHT"
100 PRINT "   ";B1$:PRINT "  ";B2$
110 PRINT:PRINT:PRINT
120 PRINT "ENTER MONTH AND DAY FOR THIS PRINTOUT"
122 INPUT "  (E.G. 7,4)";M,DA
130 PRINT:INPUT "DO YOU WANT A HARD COPY";LP$
140 PRINT:PRINT
150 IF LP$<>"Y" THEN 200
160 FOR J=1 TO 5:PRINT#130,"":NEXT J
170 PRINT#130,"";SPC(21)B1$:PRINT#130,"";SPC(20)B2$:PRINT#130,"":PRINT#130,""
180 PRINT#130,"     THIS PRINTOUT IS FOR THE FOLLOWING DATE --";
182 PRINT#130,"  ";M;"/";DA:PRINT#130,""
190 PRINT#130,"":PRINT#130,B3$:PRINT#130,""
192 PRINT#130,"":PRINT#130,F$:PRINT#130,""
200 RAD=57.2958:P1=3.14159/26
210 DIM N(12)
220 FOR I=1 TO 12
230 READ N(I)
240 NEXT I
250 DATA 0,31,59,90,120,151,181,212,243,273,304,334
260 REM EXTRA DAY IN LEAP YEARS IGNORED
270 PRINT "     ALL TIMES ARE IN UTC; SOME ARE"
272 PRINT "       NEXT (OR PREVIOUS) UTC DAY":PRINT
300 A$= "NO SUNSET TODAY"
310 B$= "NO SUNRISE TODAY"
320 PRINT "PREFIX    LOCATION   RISE  SET  HRS SUN":PRINT:LN=18
```

```
330 OPEN3,8,3,"LATLONG,S,R"
340 X=PI*(N(M)+DA)/7
350 D=.456-22.915*COS(X)-.43*COS(2*X)-.156*COS(3*X)
352 D=D+3.83*SIN(X)+.06*SIN(2*X)-.082*SIN(3*X)
360 E=.008+.51*COS(X)-3.197*COS(2*X)-.106*COS(3*X)-.15*COS(4*X)
362 E=E-7.317*SIN(X)-9.471*SIN(2*X)-.391*SIN(3*X)-.242*SIN(4*X)
370 REM D IS DECLINATION OF SUN IN DEGREES;
372 REM E IS EQUATION OF TIME IN MINUTES
380 DE=D/RAD
400 LN=LN+1
410 IF LN=60 AND LP$="Y" THEN GOSUB 780
420 INPUT#3,PX$,Q$,LA,LO
425 IF PX$="ZZZ" THEN 750
430 LA=LA/RAD
440 Y=TAN(DE)*TAN(LA)
450  IF Y=>1 THEN 690
460  IF Y<=-1 THEN 720
470 AR=-ATN(Y/SQR(-Y*Y+1))+1.5708
480 AC=AR*RAD
490 RI=LO/15+AC/15:RI=RI-E/60
500 SE=LO/15-AC/15:SE=SE-E/60
510 LT=SE-RI
520 IF LT<0 THEN LT=24+LT
530 IF (LT-INT(LT))>.99 THEN LT=(INT((LT*100)+.6))/100
540 LH=INT(LT)
550 LM=(LT-INT(LT))*60
560 IF RI>24 THEN RI=RI-24
570 IF RI<0 THEN RI=RI+24
580 IF (RI-INT(RI))>.99 THEN RI=(INT((RI*100)+.6))/100
590 R1=INT(RI)
600 R2=(RI-INT(RI))*60
610 IF SE>24 THEN SE=SE-24
620 IF SE<0 THEN SE=SE+24
630 IF (SE-INT(SE))>.99 THEN SE=(INT((SE*100)+.6))/100
640 S1=INT(SE)
650 S2=(SE-INT(SE))*60
651 R2=INT(R2+.5):S2=INT(S2+.5):LM=INT(LM+.5):D=INT(D*100+.5)/100
652 R1$=STR$(R1):R2$=STR$(R2):S1$=STR$(S1):S2$=STR$(S2)
653 LH$=STR$(LH):LM$=STR$(LM)
654 IF R1<10 THEN R1$="0"+RIGHT$(R1$,1)
655 IF R2<10 THEN R2$="0"+RIGHT$(R2$,1)
656 IF S1<10 THEN S1$="0"+RIGHT$(S1$,1)
657 IF S2<10 THEN S2$="0"+RIGHT$(S2$,1)
658 IF LH<10 THEN LH$="0"+RIGHT$(LH$,1)
659 IF LM<10 THEN LM$="0"+RIGHT$(LM$,1)
660 RX$=RIGHT$(R1$,2)+":"+RIGHT$(R2$,2):SX$=RIGHT$(S1$,2)+":"+RIGHT$(S2$,2)
661 LX$=RIGHT$(LH$,2)+":"+RIGHT$(LM$,2)
662 PRINT LEFT$(PX$,5);TAB(6)LEFT$(Q$,13);TAB(20)RX$;TAB(27)SX$;TAB(34)LX$
670 IF LP$<>"Y" THEN 680
672 PRINT#130,"    ";PX$;SPC(9-LEN(PX$))Q$;SPC(30-LEN(Q$))RX$;
674 PRINT#130,"";SPC(6)SX$;SPC(8)LX$
680 GOTO 400
690 PRINT LEFT$(PX$,5);TAB(6)LEFT$(Q$,13);TAB(20)A$
700 IF LP$="Y" THEN PRINT#130,"";SPC(4)PX$;SPC(9-LEN(PX$))Q$;SPC(30-LEN(Q$))A$
710 GOTO 400
720 PRINT LEFT$(PX$,5);TAB(6)LEFT$(Q$,13);TAB(20)B$
730 IF LP$="Y" THEN PRINT#130,"";SPC(4)PX$;SPC(9-LEN(PX$))Q$;SPC(30-LEN(Q$))B$
740 GOTO 400
750 CLOSE3
760 FOR J=LN TO 65:PRINT#130,"":NEXT J
770 FOR J=1 TO 1999:PRINT;:NEXT J:CLOSE130:CLOSE15:END
780 FOR J=1 TO 11:PRINT#130,"":NEXT J
790 PRINT#130,F$:PRINT#130,"":LN=7
800 RETURN
```

THE GRAYLINE PROGRAM

What it does

The GRAYLINE program, as the name suggests, determines who is on your gray line on any given day. It computes the sunrise and sunset times for all locations in the LATLONG data file and compares the times with your local sunrise and sunset times. After comparing all of the sunrise/sunset times and then sorting those close to yours into chronological order, it displays and prints the results.

You can specify a spread of up to 45 minutes either side of your own sunrise or sunset times. The program will print out two separate lists: one of places having their sunrise or sunset within the specified number of minutes of your sunrise, and another of places having their sunrise or sunset within the same time spread of your sunset.

For non-DXers, the term *gray line* deserves some explanation. Its origin is not entirely clear, but apparently the term was first used in print by Dale Hoppe (K6UA) in an article in *CQ Magazine* in the mid-1970s. As low-frequency DXers know so well, when both ends of a path are near sunrise or sunset simultaneously (i.e., they're both on the gray line at once), dramatic things can happen. The judicious use of this program may produce some crucial extra multipliers on the lower bands in DX contests.

One of the interesting things that can be done with this program is to compare the gray line at widely separated times throughout the year. The seasonal variations may surprise you. Places far to the east or west of your location will show up on your "gray line" at certain seasons, provided they also happen to be considerably farther north or south than you are. And it's fascinating to think who might be standing on a beach half a world away watching the sun set just when ole Sol is helping you start your day.

```
GRAY LINE CALCULATION PROGRAM

WHAT IS YOUR CALL?N6NB

DO YOU WANT A HARD COPY? Y

ENTER MONTH AND DAY
FOR THIS PRINTOUT (E.G. 7,4)10,25

HOW WIDE A GRAY LINE DO YOU WANT?
THAT IS, HOW CLOSE TO YOUR TIMES MUST
A SUNRISE OR SUNSET TIME BE FOR
LISTING BY THIS PROGRAM?
ENTER # MINS (45 MAX) 15

ENTER YOUR COORDINATES IN DEGREES, MINS
USE '-' FOR BOTH DEGREES AND MINS
IF EAST LONGITUDE, SOUTH LATITUDE

LATITUDE (E.G. 33,30)44,17
LONGITUDE (E.G. 116,30)
```

The GRAYLINE program computes the sunrise and sunset times at more than 400 locations around the world and compares those times with your local sunrise and sunset times. To run the program, you answer the questions shown in this display: your latitude and longitude, the date, the "width" of the desired gray line calculation, and so on.

Setup instructions

The GRAYLINE program requires no special modifications to work on most systems. However, take note of the point made regarding the SUNRISE CHART program: all of these sunrise/sunset-based programs calculate *actual* times, not the *apparent* sunrise and sunset times.

As explained before, you can compute apparent rather than actual sunrise times by adding and subtracting a few minutes from the computed times, but it isn't recommended for this particular program.

If you have a special interest in gray line calculations for certain specific locations around the world, by all means include them in the LATLONG data file. As we noted earlier, though, if you go over 500 total entries you have to change certain DIM (dimension) statements (see line 120).

Operating notes

To run the GRAYLINE program, enter your call, the date, and your own latitude and longitude (again, use negative numbers for BOTH degrees and minutes in the case of east longitude or south latitude). Then the crucial entry: the time spread. If you want a narrow gray line, specify a small value, such as 10 or 15 minutes. Then only places having a sunrise or sunset at almost the same time as yours will be listed. If you go to the maximum spread allowed (45 minutes), you'll probably get a list a couple of pages long.

Your answer to the gray line time spread question will depend largely on your location and purposes. On 80 meters and especially 160 meters, propagation enhancement is likely to occur only at times much closer to sunrise or sunset than is the case on 40 meters. And at locations near the equator, the period of twilight is shorter than it is at more northerly and southerly latitudes. It's not unusual, in fact, for there to be twilight all night long in the summertime near the Arctic Circle.

However, this program is written to compute the gray line only for places that actually have sunrises and sunsets, not for locations so near the poles that they are in perpetual summertime sunlight and wintertime darkness. The GRAYLINE program only works near the poles during the few weeks in the spring and fall when there actually are "days" in the usual sense at those latitudes. If you specify a latitude at which there is no sunrise or sunset, the program will advise you of this fact and stop.

Of necessity, the GRAYLINE program will give your computer a workout—and you'll get to see something of its computing speed. On most personal computers we've tried, it takes about two minutes to read in the entire LATLONG data file from disk, and about ten minutes to compute the sunrise and sunset times for all of those places. Then it may take anywhere from a few seconds to several minutes to chronologically sort the sunrise/sunset times of places that fall on your gray line, depending on the time range you specify. If your computer seems to be "stuck" and doesn't look like it's doing anything, *don't* panic and reboot it. This all takes time.

You may want to have a cup of coffee while your computer is crunching all these numbers. But if it takes your computer 15 minutes to do all of this work, imagine how long it would take if you had to do all of this by hand—or even with a pocket calculator. You can bet this isn't

something you'd do very often if you didn't own a computer! You might use a map and a rule to look at the gray line, but you wouldn't even consider actually calculating the sunrise and sunset times all over the world and then matching them up without a computer.

If waiting 15 minutes for your computer to perform thousands of mathematical calculations is annoying, there's a solution: a BASIC compiler. As we indicated in Chapter 5, on an Apple II the running time for this program can be cut down to less than four minutes with the Hayden Applesoft Compiler. This is just the kind of program on which a compiler is most impressive. For CP/M and TRS-80 users, the Microsoft Compiler will produce similarly impressive results.

Modifications for CP/M

The TRS-80 and CP/M listing should be modified as follows for those running BASIC-80 on CP/M systems:

1. Delete the CLEAR statement in line 45;
2. Replace the clear-screen command in line 80 with the proper one for your system.

Partial sample printout of GRAYLINE program.

```
            GRAY LINE CALCULATION PROGRAM

FOR N6NB AT 33.83 N LAT  117.75 W LONG


YOUR SUNRISE OCCURS AT 15:0 GMT
YOUR SUNSET OCCURS AT  0:48 GMT
ON 1/1

  - PLACES ON YOUR GRAY LINE TODAY -

AT YOUR SUNRISE...

TIME (GMT)                LOCATION

14:31         (SET)   - YK            SYRIA,DAMASCUS
14:33         (SET)   - FB8Y          ANTARCTICA,DUMON
14:33         (RISE)  - VE4           MAN,WINNIPEG
14:34         (RISE)  - W0            ND,BISMARCK
14:35         (SET)   - 3B6,7         ST BRANDON & AGA
14:35         (SET)   - OD            LEBANON,BEIRUT
14:37         (RISE)  - W7            ARIZ,PHOENIX
14:38         (SET)   - JY            JORDAN,AMMAN
14:39         (SET)   - OZ            DENMARK,COPENHAG
14:40         (SET)   - 7O            S YEMEN
14:41         (SET)   - YO            ROMANIA,BUCHARES
14:41         (SET)   - 4W            YEMEN
14:42         (SET)   - 4X,4Z         ISRAEL
14:42         (SET)   - 5B,ZC         CYPRUS
14:50         (SET)   - 3B8           MAURITIUS
```

GRAYLINE (TRS-80 and CP/M)

```
10 REM GRAY LINE DX CALCULATOR PROGRAM (TRS-80)
20 REM --LISTS ALL LOCATIONS HAVING SUNRISE/SUNSET TIMES
30 REM AT ABOUT THE SAME TIME AS YOURS.
40 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
45 CLEAR 20000
50 B1$="GRAY LINE CALCULATION PROGRAM"
60 REM BASED ON ACTUAL SUNRISE/SUNSET TIMES; APPARENT TIMES VARY
```

```
70 REM SLIGHTLY DUE TO ATMOSPHERIC REFRACTION.
80 CLS:REM CLEAR SCREEN
90 DEF FNA(X)=INT(100*X+.5)/100
100 DIM R2(100):DIM R3$(100):DIM R4$(100)
110 DIM S2(100):DIM S3$(100):DIM S4$(100)
120 DIM PX$(500):DIM Q$(500):DIM AX(500):DIM OX(500)
130 REM THE DATA FILE (LATLONG) MUST BE ON LOGGED DISK.
140 P1=3.14159/26:RAD=57.2958:CT=1:W=1:U=1:QQ=0
150 PRINT:PRINT:PRINT
160 PRINT TAB(14)B1$
165 PRINT TAB(14)STRING$(29,131)
170 KL$="":PRINT:INPUT "WHAT IS YOUR CALL?";KL$
180 LP$="":PRINT:INPUT "DO YOU WANT A HARD COPY";LP$
190 PRINT:INPUT "ENTER MONTH AND DAY FOR THIS PRINTOUT (E.G. 7,4)";M,DA
200 PRINT:PRINT "HOW WIDE A GRAY LINE DO YOU WANT?"
210 PRINT "THAT IS, HOW CLOSE TO YOUR TIMES MUST A SUNRISE/SUNSET TIME BE"
220 INPUT "FOR LISTING BY THIS PROGRAM?  ENTER # MINS (45 MAX)";GL
230 IF GL>45 THEN PRINT "TOO LONG.  45 MINS MAXIMUM":GOTO 210
240 PRINT:PRINT "ENTER YOUR LATITUDE AND LONGITUDE IN DEGREES, MINS"
250 PRINT "USE '-' FOR BOTH DEGREES AND MINS"
252 PRINT "IF EAST LONGITUDE, SOUTH LATITUDE"
260 INPUT "LATITUDE (E.G. 33,30)";LA,AM
270 INPUT "LONGITUDE (E.G. 116,30)";LO,OM
280 LA=FNA(LA+AM/60):LO=FNA(LO+OM/60)
290 A3=LA/RAD:GL=GL/60:R$="- SUNRISE AT":S$="- SUNSET AT "
300 IF LA=>0 THEN A5$=" N LAT; " ELSE A5$="S LAT; "
310 IF LO=>0 THEN O5$=" W LONG" ELSE O5$=" E LONG"
320 F1$="HERE ARE THE PLACES ON YOUR GRAY LINE TODAY"
330 F2$="TIME (GMT)                    LOCATION"
340 F3$="AT YOUR SUNRISE...":F4$="AT YOUR SUNSET..."
350 PRINT:PRINT "COMPUTING LOCAL SUNRISE DATA..."
360 GOSUB 1300:GOSUB 1430
370 IF ZZ=1 THEN 1580
380 IF ZZ=2 THEN 1590
390 LR=RI:LS=ST
400 IF LR+GL>24 THEN QQ=1
410 IF LS+GL>24 THEN QQ=2
420 IF LR-GL<0 THEN QQ=3
430 IF LS-GL<0 THEN QQ=4
440 GOSUB 1610
450 IF LP$="Y" THEN GOSUB 740
460 PRINT:PRINT "AT ";ABS(LA);A5$;" ";ABS(LO);O5$
470 PRINT:PRINT "ON THIS DATE: ";M;"/";DA
480 PRINT:PRINT "YOUR SUNRISE OCCURS AT ";RI$;" GMT"
490 PRINT "YOUR SUNSET OCCURS AT  ";ST$;" GMT"
500 REM READING ALL DATA INTO ARRAYS (SAVES DISK ACCESS TIME)
510 PRINT:PRINT "NOW READING IN DATA..."
520 OPEN "I",1,"LATLONG"
550 INPUT #1,PX$(CT),Q$(CT),AX(CT),OX(CT)
560 IF PX$(CT)="ZZZ" THEN PX$(CT)="":Q$(CT)="":GOTO 600
570 Q$(CT)=LEFT$(Q$(CT),20)
580 CT=CT+1
590 GOTO 550
600 CLOSE
610 PRINT:PRINT "NOW COMPUTING SUNRISE AND SUNSET TIMES FOR:"
620 FOR J=1 TO CT
630 A3=AX(J)/RAD:LO=OX(J)
640 GOSUB 1430
650 IF ZZ>0 THEN 710
660 IF QQ<>0 THEN GOSUB 1690
670 IF RI<LR+GL AND RI>LR-GL THEN GOSUB 830
680 IF RI<LS+GL AND RI>LS-GL THEN GOSUB 870
690 IF ST<LR+GL AND ST>LR-GL THEN GOSUB 840
700 IF ST<LS+GL AND ST>LS-GL THEN GOSUB 880
710 PX$(J)="":Q$(J)=""
720 NEXT J
730 GOTO 910
740 LPRINT:LPRINT:LPRINT TAB(12)B1$:LPRINT
750 LPRINT TAB(5)"FOR ";KL$;" AT";ABS(LA);A5$;" ";ABS(LO);O5$
760 LPRINT:LPRINT:LPRINT TAB(5)"YOUR SUNRISE OCCURS AT ";RI$;" GMT"
```

```
770 LPRINT TAB(5)"YOUR SUNSET OCCURS AT  ";ST$;" GMT"
780 LPRINT TAB(5)"ON ";M;"/";DA
790 LPRINT:LPRINT TAB(5)F1$
800 LPRINT:LPRINT TAB(5)F3$:LPRINT:LPRINT TAB(5)F2$:LPRINT:LN=17
810 RETURN
820 REM ARRAY OF PLACES ON GRAY LINE AT LOCAL SUNRISE
830 R2(W)=RI:R3$(W)=R$+" - "+PX$(J):R4$(W)=Q$(J):GOTO 850
840 R2(W)=ST:R3$(W)=S$+" - "+PX$(J):R4$(W)=Q$(J)
850 W=W+1:RETURN
860 REM ARRAY OF PLACES ON GRAY LINE AT LOCAL SUNSET
870 S2(U)=RI:S3$(U)=R$+" - "+PX$(J):S4$(U)=Q$(J):GOTO 890
880 S2(U)=ST:S3$(U)=S$+" - "+PX$(J):S4$(U)=Q$(J)
890 U=U+1:RETURN
900 REM BUBBLE SORTING EACH ARRAY...
910 PRINT:PRINT "NOW SORTING..."
920 W=W-1:U=U-1
930 FOR A1=1 TO W-1
940 F=0
950 FOR H=1 TO W-A1
960 IF R2(H+1)=>R2(H) THEN 1010
970 T=R2(H):R2(H)=R2(H+1):R2(H+1)=T
980 T$=R3$(H):R3$(H)=R3$(H+1):R3$(H+1)=T$
990 T$=R4$(H):R4$(H)=R4$(H+1):R4$(H+1)=T$
1000 F=1
1010 NEXT H
1020 IF F=0 THEN 1040
1030 NEXT A1
1040 PRINT:PRINT F1$:PRINT:PRINT F3$:PRINT:PRINT F2$
1050 IF QQ<>0 THEN 1780
1060 FOR J=1 TO W
1070 GOSUB 1920
1080 NEXT J
1090 IF LP$="Y" THEN LPRINT:LPRINT:LPRINT TAB(5)F4$
1100 IF LP$="Y" THEN LPRINT:LPRINT TAB(5)F2$:LPRINT:LN=LN+6
1110 PRINT:PRINT "NOW SORTING AGAIN..."
1120 FOR A1=1 TO U-1
1130 F=0
1140 FOR H=1 TO U-A1
1150 IF S2(H+1)=>S2(H) THEN 1200
1160 T=S2(H):S2(H)=S2(H+1):S2(H+1)=T
1170 T$=S3$(H):S3$(H)=S3$(H+1):S3$(H+1)=T$
1180 T$=S4$(H):S4$(H)=S4$(H+1):S4$(H+1)=T$
1190 F=1
1200 NEXT H
1210 IF F=0 THEN 1230
1220 NEXT A1
1230 PRINT:PRINT F4$:PRINT:PRINT F2$
1240 IF QQ<>0 THEN 1850
1250 FOR J=1 TO U
1260 GOSUB 1980
1270 NEXT J
1280 END
1290 REM COMPUTATION OF DECL, EQUATION OF TIME FOR SPECIFIC DAY
1300 DIM N(12)
1310 FOR I=1 TO 12
1320 READ N(I)
1330 NEXT I
1340 DATA 0,31,59,90,120,151,181,212,243,273,304,334
1350 REM EXTRA DAY IN LEAP YEARS IGNORED
1360 X=P1*(N(M)+DA)/7
1370 D=.456-22.915*COS(X)-.43*COS(2*X)-.156*COS(3*X)+3.83*SIN(X)+.06*SIN(2*X)-.0
     82*SIN(3*X)
1380 E=.008+.51*COS(X)-3.197*COS(2*X)-.106*COS(3*X)-.15*COS(4*X)-7.317*SIN(X)-9.
     471*SIN(2*X)-.391*SIN(3*X)-.242*SIN(4*X)
1390 REM D IS DECLINATION OF SUN IN DEGREES; E IS EQUATION OF TIME IN MINUTES
1400 DE=D/RAD
1410 RETURN
1420 REM COMPUTATION OF SUNRISE AND SUNSET TIMES
1430 Y=TAN(DE)*TAN(A3)
```

```
1440 IF Y=>1 THEN ZZ=1:RETURN
1450 IF Y<=-1 THEN ZZ=2:RETURN
1460 AR=-ATN(Y/SQR(-Y*Y+1))+1.5707633#
1470 AC=AR*RAD
1480 RI=LO/15+AC/15:RI=RI-E/60
1490 ST=LO/15-AC/15:ST=ST-E/60
1500 IF RI>24 THEN RI=RI-24
1510 IF RI<0 THEN RI=RI+24
1520 IF (RI-INT(RI))>.99 THEN RI=(INT((RI*100)+.6))/100
1530 IF ST>24 THEN ST=ST-24
1540 IF ST<0 THEN ST=ST+24
1550 IF (ST-INT(ST))>.99 THEN ST=(INT((ST*100)+.6))/100
1560 PRINT Q$(J):ZZ=0
1570 RETURN
1580 PRINT "NO SUNSET AT YOUR LOCATION TODAY":GOTO 1600
1590 PRINT "NO SUNRISE AT YOUR LOCATION TODAY"
1600 END
1610 R1=INT(RI):R2=(RI-INT(RI))*60:R2=INT(R2+.5)
1620 S1=INT(ST):S2=(ST-INT(ST))*60:S2=INT(S2+.5)
1630 RI$=STR$(R1)+":"+STR$(R2):ST$=STR$(S1)+":"+STR$(S2)
1640 RETURN
1650 T1=INT(TM):T2=(TM-INT(TM))*60:T2=INT(T2+.5)
1660 TM$=STR$(T1)+":"
1670 IF T2<10 THEN TM$=STR$(T1)+": "
1680 RETURN
1690 IF QQ=1 AND RI+24<LR+GL THEN GOSUB 830
1700 IF QQ=1 AND ST+24<LR+GL THEN GOSUB 840
1710 IF QQ=2 AND RI+24<LS+GL THEN GOSUB 870
1720 IF QQ=2 AND ST+24<LS+GL THEN GOSUB 880
1730 IF QQ=3 AND RI-24>LR-GL THEN GOSUB 830
1740 IF QQ=3 AND ST-24>LR-GL THEN GOSUB 840
1750 IF QQ=4 AND RI-24>LS-GL THEN GOSUB 870
1760 IF QQ=4 AND ST-24>LS-GL THEN GOSUB 880
1770 RETURN
1780 FOR J=1 TO W
1790 IF R2(J)>22 THEN GOSUB 1920
1800 NEXT J
1810 FOR J=1 TO W
1820 IF R2(J)<22 THEN GOSUB 1920
1830 NEXT J
1840 GOTO 1090
1850 FOR J=1 TO U
1860 IF S2(J)>22 THEN GOSUB 1980
1870 NEXT J
1880 FOR J=1 TO U
1890 IF S2(J)<22 THEN GOSUB 1980
1900 NEXT J
1910 END
1920 TM=R2(J):GOSUB 1650
1930 PRINT TM$;T2;TAB(9)R3$(J);TAB(32)R4$(J)
1940 IF LP$="Y" THEN LPRINT TAB(5)TM$;T2;TAB(13)R3$(J);TAB(37)R4$(J)
1950 IF LP$="Y" THEN LN=LN+1
1960 IF LN=>55 THEN LPRINT CHR$(12):LN=5
1970 RETURN
1980 TM=S2(J):GOSUB 1650
1990 PRINT TM$;T2;TAB(9)S3$(J);TAB(32)S4$(J)
2000 IF LP$="Y" THEN LPRINT TAB(5)TM$;T2;TAB(13)S3$(J);TAB(37)S4$(J)
2010 IF LP$="Y" THEN LN=LN+1
2020 IF LN=>55 THEN LPRINT CHR$(12):LN=5
2030 RETURN
```

GRAYLINE (Apple)

```
10  REM   GRAY LINE DX CALCULATOR PROGRAM (APPLE)
20  REM   --LISTS ALL LOCATIONS HAVING SUNRISE/SUNSET TIMES WITHIN
30  REM   A SPECIFIED TIME OF YOURS.
40  REM   BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
50 B1$ = "   GRAY LINE CALCULATION PROGRAM   "
60  REM   BASED ON ACTUAL SUNRISE/SUNSET TIMES; APPARENT TIMES VARY
70  REM   SLIGHTLY DUE TO ATMOSPHERIC REFRACTION.
80  HOME : REM   CLEAR SCREEN
90  DEF  FN A(X) =  INT (100 * X + .5) / 100
100  DIM R2(100): DIM R3$(100): DIM R4$(100)
110  DIM S2(100): DIM S3$(100): DIM S4$(100)
120  DIM PX$(500): DIM Q$(500): DIM AX(500): DIM OX(500)
130  REM   'LATLONG DATA' MUST BE ON LOGGED DISK.
140 P1 = 3.14159 / 26:RAD = 57.2958:CT = 0:W = 1:U = 1:QQ = 0
142 D$ =  CHR$ (4):PR$ = D$ + "PR#1":SC$ = D$ + "PR#0"
150  PRINT : PRINT : PRINT
155  INVERSE : PRINT "                                   "
160  PRINT B1$
165  PRINT "                                   "
167  NORMAL
170  PRINT : INPUT "WHAT IS YOUR CALL?";KL$
180  PRINT : INPUT "DO YOU WANT A HARD COPY? ";LP$
190  PRINT : PRINT "ENTER MONTH AND DAY"
192  INPUT "FOR THIS PRINTOUT (E.G. 7,4)";M,DA
200  PRINT : PRINT "HOW WIDE A GRAY LINE DO YOU WANT?"
210  PRINT "THAT IS, HOW CLOSE TO YOUR TIMES MUST"
212  PRINT "A SUNRISE OR SUNSET TIME BE FOR"
220  PRINT "LISTING BY THIS PROGRAM? ": INPUT "ENTER # MINS (45 MAX) ";GL
230  IF GL > 45 THEN  PRINT "TOO LONG.  45 MINS MAXIMUM": GOTO 210
240  PRINT : PRINT "ENTER YOUR COORDINATES IN DEGREES, MINS"
250  PRINT "USE '-' FOR BOTH DEGREES AND MINS"
252  PRINT "IF EAST LONGITUDE, SOUTH LATITUDE": PRINT
260  INPUT "LATITUDE (E.G. 33,30)";LA,AM
270  INPUT "LONGITUDE (E.G. 116,30)";LO,OM
280 LA =  FN A(LA + AM / 60):LO =  FN A(LO + OM / 60)
290 A3 = LA / RAD:GL = GL / 60:R$ = "(RISE)":S$ = "(SET) "
298 A5$ = " N LAT "
300  IF LA < 0 THEN A5$ = "S LAT "
308 O5$ = " W LONG"
310  IF LO < 0 THEN O5$ = " E LONG"
320 F1$ = "  - PLACES ON YOUR GRAY LINE TODAY -"
330 F2$ = "TIME (GMT)                LOCATION"
340 F3$ = "AT YOUR SUNRISE...":F4$ = "AT YOUR SUNSET..."
350  PRINT : PRINT "COMPUTING LOCAL SUNRISE DATA..."
360  GOSUB 1300: GOSUB 1430
370  IF ZZ = 1 THEN 1580
380  IF ZZ = 2 THEN 1590
390 LR = RI:LS = ST
400  IF LR + GL > 24 THEN QQ = 1
410  IF LS + GL > 24 THEN QQ = 2
420  IF LR - GL < 0 THEN QQ = 3
430  IF LS - GL < 0 THEN QQ = 4
440  GOSUB 1610
450  IF LP$ = "Y" THEN  GOSUB 740: HOME
460  HOME : PRINT : PRINT "AT "; ABS (LA);A5$;" "; ABS (LO);O5$
470  PRINT : PRINT "ON THIS DATE: ";M;"/";DA
480  PRINT : PRINT "YOUR SUNRISE OCCURS AT ";RI$;" GMT"
490  PRINT "YOUR SUNSET OCCURS AT  ";ST$;" GMT"
500  REM   READING ALL DATA INTO ARRAYS (SAVES DISK ACCESS TIME)
510  PRINT : PRINT "NOW READING IN DATA..."
520  PRINT D$;"OPEN LATLONG DATA"
540  PRINT D$;"READ LATLONG DATA"
550  INPUT PX$(CT),Q$(CT),AX(CT),OX(CT)
560  IF PX$(CT) = "ZZZ" THEN PX$(CT) = "":Q$(CT) = "": GOTO 600
570 Q$(CT) =  LEFT$ (Q$(CT),16)
572 PX$(CT) =  LEFT$ (PX$(CT),6)
580 CT = CT + 1
```

```
590  GOTO 540
600  PRINT D$;"CLOSE"
610  PRINT : PRINT "NOW COMPUTING SUNRISE/SUNSET TIMES FOR:"
620  FOR J = 1 TO CT
630 A3 = AX(J) / RAD:LO = OX(J)
640  GOSUB 1430
650  IF ZZ > 0 THEN 710
660  IF QQ <  > 0 THEN  GOSUB 1690
670  IF RI < LR + GL AND RI > LR - GL THEN  GOSUB 830
680  IF RI < LS + GL AND RI > LS - GL THEN  GOSUB 870
690  IF ST < LR + GL AND ST > LR - GL THEN  GOSUB 840
700  IF ST < LS + GL AND ST > LS - GL THEN  GOSUB 880
710 PX$(J) = "":Q$(J) = ""
720  NEXT J
730  GOTO 910
735  REM   SEE TEXT RE APPLE PRINTER INTERFACING AND FORMATTING
740  PRINT PR$: PRINT : PRINT  TAB( 14)B1$: PRINT
750  PRINT  TAB( 5)"FOR ";KL$;" AT "; ABS (LA);A5$;" "; ABS (LO);O5$
760  PRINT : PRINT : PRINT  TAB( 5)"YOUR SUNRISE OCCURS AT ";RI$;" GMT"
770  PRINT  TAB( 5)"YOUR SUNSET OCCURS AT  ";ST$;" GMT"
780  PRINT  TAB( 5)"ON ";M;"/";DA
790  PRINT : PRINT  TAB( 5)F1$
800  PRINT : PRINT  TAB( 5)F3$: PRINT : PRINT  TAB( 5)F2$: PRINT : PRINT SC$:LN =
17
810  RETURN
820  REM   ARRAY OF PLACES ON GRAY LINE AT LOCAL SUNRISE
830 R2(W) = RI:R3$(W) = R$ + " - " + PX$(J):R4$(W) = Q$(J): GOTO 850
840 R2(W) = ST:R3$(W) = S$ + " - " + PX$(J):R4$(W) = Q$(J)
850 W = W + 1: RETURN
860  REM   ARRAY OF PLACES ON GRAY LINE AT LOCAL SUNSET
870 S2(U) = RI:S3$(U) = R$ + " - " + PX$(J):S4$(U) = Q$(J): GOTO 890
880 S2(U) = ST:S3$(U) = S$ + " - " + PX$(J):S4$(U) = Q$(J)
890 U = U + 1: RETURN
900  REM   BUBBLE SORTING EACH ARRAY...
910  PRINT : PRINT "NOW SORTING..."
920 W = W - 1:U = U - 1
930  FOR A1 = 1 TO W - 1
940 F = 0
950  FOR H = 1 TO W - A1
960  IF R2(H + 1) =  > R2(H) THEN 1010
970 T = R2(H):R2(H) = R2(H + 1):R2(H + 1) = T
980 T$ = R3$(H):R3$(H) = R3$(H + 1):R3$(H + 1) = T$
990 T$ = R4$(H):R4$(H) = R4$(H + 1):R4$(H + 1) = T$
1000 F = 1
1010  NEXT H
1020  IF F = 0 THEN 1040
1030  NEXT A1
1040  IF LP$ <  > "Y" THEN  PRINT : PRINT F1$: PRINT : PRINT F3$: PRINT : PRINT F2
$
1050  IF QQ <  > 0 THEN 1780
1060  FOR J = 1 TO W
1070  GOSUB 1920
1080  NEXT J
1090  IF LP$ = "Y" THEN  PRINT PR$: PRINT : PRINT  TAB( 5)F4$
1100  IF LP$ = "Y" THEN  PRINT : PRINT  TAB( 5)F2$: PRINT :LN = LN + 6: PRINT SC$
1110  PRINT : PRINT "NOW SORTING AGAIN..."
1120  FOR A1 = 1 TO U - 1
1130 F = 0
1140  FOR H = 1 TO U - A1
1150  IF S2(H + 1) =  > S2(H) THEN 1200
1160 T = S2(H):S2(H) = S2(H + 1):S2(H + 1) = T
1170 T$ = S3$(H):S3$(H) = S3$(H + 1):S3$(H + 1) = T$
1180 T$ = S4$(H):S4$(H) = S4$(H + 1):S4$(H + 1) = T$
1190 F = 1
1200  NEXT H
1210  IF F = 0 THEN 1230
1220  NEXT A1
1230  PRINT : PRINT F4$: PRINT : PRINT F2$
1240  IF QQ <  > 0 THEN 1850
1250  FOR J = 1 TO U
```

```
1260  GOSUB 1980
1270  NEXT J
1280  PRINT SC$: END
1290  REM   COMPUTATION OF DECL, EQUATION OF TIME FOR SPECIFIC DAY
1300  DIM N(12)
1310  FOR I = 1 TO 12
1320  READ N(I)
1330  NEXT I
1340  DATA   0,31,59,90,120,151,181,212,243,273,304,334
1350  REM   EXTRA DAY IN LEAP YEARS IGNORED
1360 X = P1 * (N(M) + DA) / 7
1370 D = .456 - 22.915 *  COS (X) - .43 *  COS (2 * X) - .156 *  COS (3 * X) + 3.8
3 *  SIN (X) + .06 *  SIN (2 * X) - .082 *  SIN (3 * X)
1380 E = .008 + .51 *  COS (X) - 3.197 *  COS (2 * X) - .106 *  COS (3 * X) - .15
*  COS (4 * X) - 7.317 *  SIN (X) - 9.471 *  SIN (2 * X) - .391 *  SIN (3 * X) - .
242 *  SIN (4 * X)
1390  REM   D IS DECLINATION OF SUN IN DEGREES; E IS EQUATION OF TIME IN MINUTES
1400 DE = D / RAD
1410  RETURN
1420  REM   COMPUTATION OF SUNRISE AND SUNSET TIMES
1430 Y =  TAN (DE) *  TAN (A3)
1440  IF Y => 1 THEN ZZ = 1: RETURN
1450  IF Y <=  - 1 THEN ZZ = 2: RETURN
1460 AR =  -  ATN (Y /  SQR ( - Y * Y + 1)) + 1.5707633
1470 AC = AR * RAD
1480 RI = LO / 15 + AC / 15:RI = RI - E / 60
1490 ST = LO / 15 - AC / 15:ST = ST - E / 60
1500  IF RI > 24 THEN RI = RI - 24
1510  IF RI < 0 THEN RI = RI + 24
1520  IF (RI -  INT (RI)) > .99 THEN RI = ( INT ((RI * 100) + .6)) / 100
1530  IF ST > 24 THEN ST = ST - 24
1540  IF ST < 0 THEN ST = ST + 24
1550  IF (ST -  INT (ST)) > .99 THEN ST = ( INT ((ST * 100) + .6)) / 100
1560  PRINT Q$(J):ZZ = 0
1570  RETURN
1580  PRINT "NO SUNSET AT YOUR LOCATION TODAY": GOTO 1600
1590  PRINT "NO SUNRISE AT YOUR LOCATION TODAY"
1600  END
1610 R1 =  INT (RI):R2 = (RI -  INT (RI)) * 60:R2 =  INT (R2 + .5)
1620 S1 =  INT (ST):S2 = (ST -  INT (ST)) * 60:S2 =  INT (S2 + .5)
1630 RI$ =  STR$ (R1) + ":" +  STR$ (R2):ST$ =  STR$ (S1) + ":" +  STR$ (S2)
1640  RETURN
1650 T1 =  INT (TM):T2 = (TM -  INT (TM)) * 60:T2 =  INT (T2 + .5)
1660 TM$ =  STR$ (T1) + ":" +  STR$ (T2)
1670  IF T2 < 10 THEN TM$ =  STR$ (T1) + ": " +  STR$ (T2)
1680  RETURN
1690  IF QQ = 1 AND RI + 24 < LR + GL THEN  GOSUB 830
1700  IF QQ = 1 AND ST + 24 < LR + GL THEN  GOSUB 840
1710  IF QQ = 2 AND RI + 24 < LS + GL THEN  GOSUB 870
1720  IF QQ = 2 AND ST + 24 < LS + GL THEN  GOSUB 880
1730  IF QQ = 3 AND RI - 24 > LR - GL THEN  GOSUB 830
1740  IF QQ = 3 AND ST - 24 > LR - GL THEN  GOSUB 840
1750  IF QQ = 4 AND RI - 24 > LS - GL THEN  GOSUB 870
1760  IF QQ = 4 AND ST - 24 > LS - GL THEN  GOSUB 880
1770  RETURN
1780  FOR J = 1 TO W
1790  IF R2(J) > 22 THEN  GOSUB 1920
1800  NEXT J
1810  FOR J = 1 TO W
1820  IF R2(J) < 22 THEN  GOSUB 1920
1830  NEXT J
1840  GOTO 1090
1850  FOR J = 1 TO U
1860  IF S2(J) > 22 THEN  GOSUB 1980
1870  NEXT J
1880  FOR J = 1 TO U
1890  IF S2(J) < 22 THEN  GOSUB 1980
1900  NEXT J
1910  END
1920 TM = R2(J): GOSUB 1650
```

```
1930  IF LP$ < > "Y" THEN  PRINT TM$; SPC( 7 -  LEN (TM$));R3$(J); SPC( 15 -  LEN
 (R3$(J)));R4$(J)
1940  IF LP$ = "Y" THEN  PRINT PR$: PRINT  TAB( 5)TM$; SPC( 13 -  LEN (TM$));R3$(J
); SPC( 23 -  LEN (R3$(J)));R4$(J)
1950  IF LP$ = "Y" THEN  PRINT SC$:LN = LN + 1
1960  IF LN >  = 55 THEN  PRINT PR$: PRINT  CHR$ (12):LN = 5: PRINT SC$
1970  RETURN
1980 TM = S2(J): GOSUB 1650
1990  IF LP$ < > "Y" THEN  PRINT TM$; SPC( 7 -  LEN (TM$));S3$(J); SPC( 15 -  LEN
 (S3$(J)));S4$(J)
2000  IF LP$ = "Y" THEN  PRINT PR$: PRINT  TAB( 5)TM$; SPC( 13 -  LEN (TM$));S3$(J
); SPC( 23 -  LEN (S3$(J)));S4$(J)
2010  IF LP$ = "Y" THEN LN = LN + 1
2020  IF LN =  > 55 THEN  PRINT PR$: PRINT  CHR$ (12):LN = 5
2030  RETURN
```

GRAYLINE (Commodore 64)

```
10 REM GRAY LINE DX CALCULATOR PROGRAM (C64)
15 OPEN 130,2,2,CHR$(136):REM RS232 PRINTER 1200B/2STOP BITS
20 REM --LISTS ALL LOCATIONS HAVING SUNRISE/SUNSET TIMES
30 REM AT ABOUT THE SAME TIME AS YOURS.
40 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
50 B1$="GRAY LINE CALCULATION PROGRAM"
60 REM BASED ON ACTUAL SUNRISE/SUNSET TIMES; APPARENT TIMES VARY
70 REM SLIGHTLY DUE TO ATMOSPHERIC REFRACTION.
80 PRINT CHR$(147): REM CLEAR SCREEN
90 DEF FNA(X)=INT(100*X+.5)/100
100 DIM R2(100):DIM R3$(100):DIM R4$(100)
110 DIM S2(100):DIM S3$(100):DIM S4$(100)
120 DIM PX$(500):DIM Q$(500):DIM AX(500):DIM OX(500)
130 REM THE DATA FILE (LATLONG) MUST BE ON LOGGED DISK.
140 P1=3.14159/26:RAD=57.2958:CT=1:W=1:U=1:QQ=0
150 PRINT:PRINT:PRINT
160 PRINT TAB(4)B1$:PRINT
170 PRINT:INPUT "WHAT IS YOUR CALL";KL$
180 PRINT:INPUT "DO YOU WANT A HARD COPY";LP$
190 PRINT:PRINT "ENTER MONTH AND DAY"
195 INPUT "FOR THIS PRINTOUT (E.G. 7,4)";M,DA
200 PRINT:PRINT "HOW WIDE A GRAY LINE DO YOU WANT?"
210 PRINT "THAT IS, HOW CLOSE TO YOUR TIMES MUST"
215 PRINT "A SUNRISE OR SUNSET TIME BE FOR"
220 PRINT "LISTING BY THIS PROGRAM?":INPUT "ENTER # MINS (45 MAX)";GL
230 IF GL>45 THEN PRINT "TOO LONG.  45 MINS MAXIMUM":GOTO 200
240 PRINT:PRINT "ENTER YOUR COORDINATES IN DEGREES, MINS"
250 PRINT "USE '-' FOR BOTH DEGREES AND MINS"
252 PRINT "IF EAST LONGITUDE, SOUTH LATITUDE":PRINT
260 INPUT "LATITUDE (E.G. 33,30)";LA,AM
270 INPUT "LONGITUDE (E.G. 116,30)";LO,OM
280 LA=FNA(LA+AM/60):LO=FNA(LO+OM/60)
290 A3=LA/RAD:GL=GL/60:R$="(RISE)":S$="(SET) "
300 A5$=" S LAT; ":IF LA=>0 THEN A5$=" N LAT; "
310 O5$=" E LONG":IF LO=>0 THEN O5$=" W LONG"
320 F1$="HERE ARE THE PLACES ON YOUR GRAY LINE TODAY"
330 F2$="TIME (GMT)              LOCATION"
340 F3$="AT YOUR SUNRISE...":F4$="AT YOUR SUNSET..."
350 PRINT:PRINT "COMPUTING LOCAL SUNRISE DATA..."
360 GOSUB 1300:GOSUB 1430
370 IF ZZ=1 THEN 1580
380 IF ZZ=2 THEN 1590
390 LR=RI:LS=SO
400 IF LR+GL>24 THEN QQ=1
410 IF LS+GL>24 THEN QQ=2
420 IF LR-GL<0 THEN QQ=3
430 IF LS-GL<0 THEN QQ=4
```

```
440 GOSUB 1610
450 IF LP$="Y" THEN GOSUB 740
460 PRINT:PRINT "AT ";ABS(LA);A5$;" ";ABS(LO);O5$
470 PRINT:PRINT "ON THIS DATE: ";M;"/";DA
480 PRINT:PRINT "YOUR SUNRISE OCCURS AT ";RI$;" GMT"
490 PRINT "YOUR SUNSET OCCURS AT  ";S0$;" GMT"
500 REM READING ALL DATA INTO ARRAYS (SAVES DISK ACCESS TIME)
510 PRINT:PRINT "NOW READING IN DATA..."
520 OPEN 8,8,8,"LATLONG,S,R"
550 INPUT#8,PX$(CT),Q$(CT),AX(CT),OX(CT)
560 IF PX$(CT)="ZZZ" THEN PX$(CT)="":Q$(CT)="":GOTO 600
570 Q$(CT)=LEFT$(Q$(CT),16)
580 CT=CT+1
590 GOTO 550
600 CLOSE8
610 PRINT:PRINT "COMPUTING SUNRISE AND SUNSET TIMES FOR:"
620 FOR J=1 TO CT
630 A3=AX(J)/RAD:LO=OX(J)
640 GOSUB 1430
650 IF ZZ>0 THEN 710
660 IF QQ<>0 THEN GOSUB 1690
670 IF RI<LR+GL AND RI>LR-GL THEN GOSUB 830
680 IF RI<LS+GL AND RI>LS-GL THEN GOSUB 870
690 IF S0<LR+GL AND S0>LR-GL THEN GOSUB 840
700 IF S0<LS+GL AND S0>LS-GL THEN GOSUB 880
710 PX$(J)="":Q$(J)=""
720 NEXT J
730 GOTO 910
740 CMD130:PRINT:PRINT:PRINT TAB(12)B1$:PRINT
750 PRINT TAB(5)"FOR ";KL$;" AT";ABS(LA);A5$;" ";ABS(LO);O5$
760 PRINT:PRINT:PRINT TAB(5)"YOUR SUNRISE OCCURS AT ";RI$;" GMT"
770 PRINT TAB(5)"YOUR SUNSET OCCURS AT  ";S0$;" GMT"
780 PRINT TAB(5)"ON ";M;"/";DA
790 PRINT:PRINT TAB(5)F1$
800 PRINT:PRINT TAB(5)F3$:PRINT:PRINT TAB(5)F2$:PRINT#130,"":LN=17
810 RETURN
820 REM ARRAY OF PLACES ON GRAY LINE AT LOCAL SUNRISE
830 R2(W)=RI:R3$(W)=R$+" - "+PX$(J):R4$(W)=Q$(J):GOTO 850
840 R2(W)=S0:R3$(W)=S$+" - "+PX$(J):R4$(W)=Q$(J)
850 W=W+1:RETURN
860 REM ARRAY OF PLACES ON GRAY LINE AT LOCAL SUNSET
870 S2(U)=RI:S3$(U)=R$+" - "+PX$(J):S4$(U)=Q$(J):GOTO 890
880 S2(U)=S0:S3$(U)=S$+" - "+PX$(J):S4$(U)=Q$(J)
890 U=U+1:RETURN
900 REM BUBBLE SORTING EACH ARRAY...
910 PRINT:PRINT "NOW SORTING..."
920 W=W-1:U=U-1
930 FOR A1=1 TO W-1
940 F=0
950 FOR H=1 TO W-A1
960 IF R2(H+1)=>R2(H) THEN 1010
970 T=R2(H):R2(H)=R2(H+1):R2(H+1)=T
980 T$=R3$(H):R3$(H)=R3$(H+1):R3$(H+1)=T$
990 T$=R4$(H):R4$(H)=R4$(H+1):R4$(H+1)=T$
1000 F=1
1010 NEXT H
1020 IF F=0 THEN 1040
1030 NEXT A1
1040 PRINT:PRINT F1$:PRINT:PRINT F3$:PRINT:PRINT F2$
1050 IF QQ<>0 THEN 1780
1060 FOR J=1 TO W
1070 GOSUB 1920
1080 NEXT J
1090 IF LP$="Y" THEN CMD130:PRINT:PRINT:PRINT TAB(5)F4$
1100 IF LP$="Y" THEN PRINT:PRINT TAB(5)F2$:PRINT#130:LN=LN+6
1110 PRINT:PRINT "NOW SORTING AGAIN..."
1120 FOR A1=1 TO U-1
1130 F=0
1140 FOR H=1 TO U-A1
```

```
1150 IF S2(H+1)=>S2(H) THEN 1200
1160 T=S2(H):S2(H)=S2(H+1):S2(H+1)=T
1170 T$=S3$(H):S3$(H)=S3$(H+1):S3$(H+1)=T$
1180 T$=S4$(H):S4$(H)=S4$(H+1):S4$(H+1)=T$
1190 F=1
1200 NEXT H
1210 IF F=0 THEN 1230
1220 NEXT A1
1230 PRINT:PRINT F4$:PRINT:PRINT F2$
1240 IF QQ<>0 THEN 1850
1250 FOR J=1 TO U
1260 GOSUB 1980
1270 NEXT J
1280 FOR J=1 TO 1000:PRINT;:NEXT J:CLOSE130:END
1290 REM COMPUTATION OF DECL, EQUATION OF TIME FOR SPECIFIC DAY
1300 DIM N(12)
1310 FOR I=1 TO 12
1320 READ N(I)
1330 NEXT I
1340 DATA 0,31,59,90,120,151,181,212,243,273,304,334
1350 REM EXTRA DAY IN LEAP YEARS IGNORED
1360 X=P1*(N(M)+DA)/7
1370 D=.456-22.915*COS(X)-.43*COS(2*X)-.156*COS(3*X)
1375 D=D+3.83*SIN(X)+.06*SIN(2*X)-.082*SIN(3*X)
1380 E=.008+.51*COS(X)-3.197*COS(2*X)-.106*COS(3*X)-.15*COS(4*X)
1385 E=E-7.317*SIN(X)-9.471*SIN(2*X)-.391*SIN(3*X)-.242*SIN(4*X)
1390 REM D IS DECLINATION OF SUN IN DEGREES; E IS EQUATION OF TIME IN MINUTES
1400 DE=D/RAD
1410 RETURN
1420 REM COMPUTATION OF SUNRISE AND SUNSET TIMES
1430 Y=TAN(DE)*TAN(A3)
1440 IF Y=>1 THEN ZZ=1:RETURN
1450 IF Y<=-1 THEN ZZ=2:RETURN
1460 AR=-ATN(Y/SQR(-Y*Y+1))+1.5707633
1470 AC=AR*RAD
1480 RI=LO/15+AC/15:RI=RI-E/60
1490 S0=LO/15-AC/15:S0=S0-E/60
1500 IF RI>24 THEN RI=RI-24
1510 IF RI<0 THEN RI=RI+24
1520 IF (RI-INT(RI))>.99 THEN RI=(INT((RI*100)+.6))/100
1530 IF S0>24 THEN S0=S0-24
1540 IF S0<0 THEN S0=S0+24
1550 IF (S0-INT(S0))>.99 THEN S0=(INT((S0*100)+.6))/100
1560 PRINT Q$(J):ZZ=0
1570 RETURN
1580 PRINT "NO SUNSET AT YOUR LOCATION TODAY":GOTO 1600
1590 PRINT "NO SUNRISE AT YOUR LOCATION TODAY"
1600 CLOSE 130:END
1610 R1=INT(RI):R2=(RI-INT(RI))*60:R2=INT(R2+.5)
1620 S1=INT(S0):S2=(S0-INT(S0))*60:S2=INT(S2+.5)
1630 RI$=STR$(R1)+":"+STR$(R2):S0$=STR$(S1)+":"+STR$(S2)
1640 RETURN
1650 T1=INT(TM):T2=(TM-INT(TM))*60:T2=INT(T2+.5)
1660 TM$=STR$(T1)+":"
1670 IF T2<10 THEN TM$=STR$(T1)+": "
1680 RETURN
1690 IF QQ=1 AND RI+24<LR+GL THEN GOSUB 830
1700 IF QQ=1 AND S0+24<LR+GL THEN GOSUB 840
1710 IF QQ=2 AND RI+24<LS+GL THEN GOSUB 870
1720 IF QQ=2 AND S0+24<LS+GL THEN GOSUB 880
1730 IF QQ=3 AND RI-24>LR-GL THEN GOSUB 830
1740 IF QQ=3 AND S0-24>LR-GL THEN GOSUB 840
1750 IF QQ=4 AND RI-24>LS-GL THEN GOSUB 870
1760 IF QQ=4 AND S0-24>LS-GL THEN GOSUB 880
1770 RETURN
1780 FOR J=1 TO W
1790 IF R2(J)>22 THEN GOSUB 1920
1800 NEXT J
1810 FOR J=1 TO W
1820 IF R2(J)<22 THEN GOSUB 1920
```

```
1830 NEXT J
1840 GOTO 1090
1850 FOR J=1 TO U
1860 IF S2(J)>22 THEN GOSUB 1980
1870 NEXT J
1880 FOR J=1 TO U
1890 IF S2(J)<22 THEN GOSUB 1980
1900 NEXT J
1910 END
1920 TM=R2(J):GOSUB 1650
1930 PRINT TM$T2;TAB(7)R3$(J);SPC(3)LEFT$(R4$(J),14)
1940 IF LP$<>"Y" THEN RETURN
1942 CMD130:PRINT TAB(5)TM$T2;SPC(5)R3$(J);SPC(24-LEN(R3$(J)))R4$(J);
1950 LN=LN+1
1960 IF LN=>55 THEN PRINT CHR$(12):LN=5
1970 PRINT#130,"";:RETURN
1980 TM=S2(J):GOSUB 1650
1990 PRINT TM$T2;TAB(7)R3$(J);SPC(3)LEFT$(R4$(J),14)
2000 IF LP$<>"Y" THEN RETURN
2002 CMD130:PRINT TAB(5)TM$T2;SPC(5)R3$(J);SPC(24-LEN(R3$(J)))R4$(J);
2010 LN=LN+1
2020 IF LN=>55 THEN PRINT CHR$(12):LN=5
2030 PRINT#130,"";:RETURN
```

THE GENERAL SUNRISE/SUNSET PROGRAM

What it does

This program calculates and prints out a table of sunrise and sunset times for any one place over a period of time. It does not require the LATLONG data file, so it is especially suited for systems without disk drives. It will also work without a printer: the data will display on the screen.

Whereas the SUNRISE CHART program prints sunrise/sunset times for many locations on one day, this program will print data for one location on any number of days (up to a full year), beginning on any day you specify. It uses the same mathematical formulas as the other sunrise-based programs, but with different data-handling, display, and print features.

Setup instructions

As with the other sunrise-based programs, this one computes the actual sunrise/sunset times, not the apparent ones. However, users of this program may be particularly interested in the apparent sunrise times, allowing for atmospheric refraction.

If you want apparent times instead of actual times, add a few minutes to the sunset times by adding about 3 to the value of S2 in line 990 on the Apple version and line 920 of the CP/M and TRS-80 listing. Likewise, for apparent sunrise times subtract 3 from the value of R2 on line 1000 in the Apple version and line 870 in the TRS-80 and CP/M listing.

Operating notes

To run this program, enter the starting date, the number of days for which you want data computed, and the longitude and latitude of the desired location.

As in the other programs involving latitude and longitude, use a minus sign for both degrees and minutes in the case of east longitude or south latitude.

Modifications for CP/M

For CP/M systems running BASIC-80, make these changes to the TRS-80 and CP/M listing:

1. Delete the CLEAR statement in line 8;
2. Change the clear-screen command in line 80 to the proper command for your system.

SUNRISES ANYWHERE (TRS-80 and CP/M)

```
5 REM SUNRISE PROGRAM FOR ANY NUMBER OF DAYS AT ANY ONE PLACE (TRS-80 VERSION)
7 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
8 CLEAR 250
10 B1$= "SUNRISE/SUNSET CALCULATION PROGRAM"
20 B2$= "FOR ANY POINT ON THE GLOBE"
50 REM NO DATA FILE IS REQUIRED.
60 REM TIMES GIVEN ARE ACTUAL SUNRISE/SUNSET.  APPARENT SUNRISE IS 5-7 MINS.
70 REM EARLIER AND SUNSET A LIKE AMOUNT LATER DUE TO ATMOSPHERIC REFRACTION.
80 CLS:REM CLEAR SCREEN
90 PRINT TAB(9)B1$:PRINT TAB(13)B2$:PRINT:PRINT:PRINT
100 INPUT "ENTER STARTING MONTH AND DAY (E.G. 7,4)";M,DA
110 PRINT
120 PRINT "ENTER COORDINATES, USING '-' FOR BOTH DEGREES AND MINS"
122 PRINT "   IF EAST LONGITUDE, SOUTH LATITUDE"
130 PRINT:INPUT "ENTER THE LATITUDE IN DEGREES, MINUTES";A1,A2
140 PRINT
150 INPUT "ENTER THE LONGITUDE IN DEGREES, MINUTES";O1,O2
160 PRINT:LINE INPUT "NAME OF THIS GEOGRAPHIC LOCATION: ";G$
170 PRINT
180 INPUT "FOR HOW MANY DAYS DO YOU WANT A PRINTOUT?";NR
190 PRINT:INPUT "DO YOU WANT A HARD COPY?";LP$
200 PRINT:PRINT:PRINT
210 IF LP$<>"Y" THEN 240
220 FOR J=1 TO 5:LPRINT:NEXT J
230 LPRINT TAB(17)B1$:LPRINT TAB(21)B2$:LPRINT:LPRINT
240 A3=A1+A2/60:O3=O1+O2/60
250 IF A3=>0 THEN A5$=" N LAT;  " ELSE A5$=" S LAT;  "
260 IF O3=>0 THEN O5$=" W LONG" ELSE O5$="E LONG"
270 A4=INT(A3*100+.5)/100
280 O4=INT(O3*100+.5)/100
290 RAD=57.2958:P1=3.14159/26
300 A3=A3/RAD
310 DIM N(12)
320 FOR I=1 TO 12
330 READ N(I)
340 NEXT I
350 DATA 0,31,59,90,120,151,181,212,243,273,304,334
360 REM EXTRA DAY IN LEAP YEARS IGNORED
370 IF LP$<>"Y" THEN 410
380 LPRINT "    THIS COMPUTATION IS FOR: ";ABS(A4);A5$;ABS(O4);O5$
390 LPRINT:LPRINT "    GEOGRAPHIC LOCATION: ";G$
400 LPRINT
410 K$= "    ALL TIMES ARE IN GMT; SOME MAY BE NEXT (OR PREVIOUS) GMT DAY"
420 PRINT K$:PRINT
430 A$="     ## ##       ## ##       ## ##     +###.##         ## ##"
440 B$= "THIS LATITUDE THE SUN DOESN'T SET ON: "
450 C$= "THIS LATITUDE THE SUN DOESN'T RISE ON: "
460 F$= "     DATE      SUNRISE      SUNSET   SUN'S DECL   TOTAL SUN"
470 J$= "    MO DAY    HRS  MINS    HRS MINS   DEGREES      HRS MINS"
480 PRINT F$:PRINT J$:PRINT
490 IF LP$<>"Y" THEN GOTO 520
500 LPRINT K$:LPRINT:LPRINT
510 LPRINT F$:LPRINT J$:LPRINT
520 K=N(M)+DA
530 LN=19
540 FOR Q=1 TO NR
550 LN=LN+1
560 IF LN=60 AND LP$="Y" THEN GOSUB 1050
```

```
570 DT = K-N(M)
580 IF DT=32 THEN DT = 1
590 IF K<366 THEN 610
600 M=1 : DA=1 : K=1
610 IF M=12 THEN 640
620 IF K>N(M+1) THEN DT=K-(N(M+1))
630 IF K>N(M+1) THEN M=M+1
640 X=P1*K/7
650 K=K+1
660 D=.456-22.915*COS(X)-.43*COS(2*X)-.156*COS(3*X)+3.83*SIN(X)+.06*SIN(2*X)-.08
    2*SIN(3*X)
670 E=.008+.51*COS(X)-3.197*COS(2*X)-.106*COS(3*X)-.15*COS(4*X)-7.317*SIN(X)-9.4
    71*SIN(2*X)-.391*SIN(3*X)-.242*SIN(4*X)
680 REM D IS DECLINATION OF SUN IN DEGREES; E IS EQUATION OF TIME IN MINUTES
690 DE=D/RAD
700 Y=TAN(DE)*TAN(A3)
710 IF A3<0 THEN H$="BELOW " ELSE H$="ABOVE "
720 IF Y=>1 THEN 960
730 IF Y<=-1 THEN 990
740 AR=-ATN(Y/SQR(-Y*Y+1))+1.5707633#
750 AC=AR*RAD
760 RI=O3/15+AC/15:RI=RI-E/60
770 ST=O3/15-AC/15:ST=ST-E/60
780 LT=ST-RI
790 IF LT<0 THEN LT=24+LT
800 IF (LT-INT(LT))>.99 THEN LT=(INT((LT*100)+.6))/100
810 LH=INT(LT)
820 LM=(LT-INT(LT))*60
830 IF RI>24 THEN RI=RI-24
840 IF RI<0 THEN RI=RI+24
850 IF (RI-INT(RI))>.99 THEN RI=(INT((RI*100)+.6))/100
860 R1=INT(RI)
870 R2=(RI-INT(RI))*60
880 IF ST>24 THEN ST=ST-24
890 IF ST<0 THEN ST=ST+24
900 IF (ST-INT(ST))>.99 THEN ST=(INT((ST*100)+.6))/100
910 S1=INT(ST)
920 S2=(ST-INT(ST))*60
930 PRINT USING A$;M;DT;R1;R2;S1;S2;D;LH;LM
940 IF LP$="Y" THEN LPRINT USING A$;M;DT;R1;R2;S1;S2;D;LH;LM
950 GOTO 1010
960 PRINT H$;B$;M;DT
970 IF LP$="Y" THEN LPRINT TAB(5)H$;B$;M;DT
980 GOTO 1010
990 PRINT H$;C$;M;DT
1000 IF LP$="Y" THEN LPRINT TAB(5)H$;C$;M;DT
1010 NEXT Q
1020 IF LP$<>"Y" THEN 1040
1030 FOR J=LN TO 65:LPRINT:NEXT J
1040 END
1050 FOR J=1 TO 11:LPRINT:NEXT J
1060 LPRINT F$:LPRINT J$:LPRINT:LN=8
1070 RETURN
```

SUNRISES ANYWHERE (Apple)

```
10  REM  SUNRISE/SUNSET PROGRAM FOR ANY NUMBER OF DAYS AT ONE PLACE
15  REM  BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
20 D$ =  CHR$ (4):I$ =  CHR$ (9):X$ = "80N":P$ = "PR#0":P1$ = "PR#1"
30  HOME
40 B1$ = "   SUNRISE/SUNSET CALCULATION PROGRAM"
50 B2$ = "         FOR ANY POINT ON THE GLOBE"
60 B3$ = "ALL TIMES ARE IN GMT; SOME MAY BE NEXT  (OR PREVIOUS)  GMT DAY."
70  PRINT B1$
80  PRINT : PRINT B2$: PRINT : PRINT
90  FOR J = 1 TO 10: PRINT : NEXT J
100  REM   NO DATA FILE IS REQUIRED FOR THIS SUNRISE/SUNSET PROGRAM;
```

```
110  REM  TIMES GIVEN ARE ACTUAL TIMES; APPARENT SUNRISE IS 5-7 MINS
114  REM  EARLIER AND SUNSET A LIKE AMOUNT LATER DUE TO ATMOSPHERIC REFACTION
120  INPUT "DO YOU WANT A HARD COPY PRINTOUT? (Y/N) ";LP$: PRINT : PRINT
130  PRINT : PRINT
180  IF LP$ = "Y" THEN  PRINT D$;P1$: PRINT I$;X$: PRINT : PRINT : PRINT  SPC( 16)
;B1$: PRINT : PRINT  SPC( 16)B2$: PRINT
190  PRINT D$;P$
200  INPUT "ENTER THE STARTING MONTH AND DAY (E.G. 7,4)";M,DA
210  PRINT
220  PRINT "ENTER THE COORDINATES, USING '-' FOR"
222  PRINT "BOTH DEGREES AND MINS IF E LAT, S LONG"
230  PRINT
240  INPUT "ENTER THE LATITUDE IN DEGREES, MINUTES  ";A1,A2
250 A3 = A1 + A2 / 60
260  IF A3 <  = 90 AND A3 >  =  - 90 GOTO 300
270  PRINT "LATITUDE MUST BE BETWEEN -90 AND +90": GOTO 240
300  INPUT "ENTER THE LONGITUDE IN DEGREES, MINUTES ";O1,O2
310 O3 = O1 + O2 / 60
312  IF O3 <  = 180 AND O3 >  =  - 180 GOTO 320
314  PRINT "LONGITUDE MUST BE BETWEEN -180 AND +180": GOTO 300
320 A4 = A3
330  PRINT
340  INPUT "NAME OF THIS GEOGRAPHIC LOCATION ";G$
350  PRINT
360  INPUT "FOR HOW MANY DAYS DO YOU WANT A PRINTOUT? ";NR
370  PRINT
380  PRINT : PRINT : PRINT
390 RA = 57.2958:P = 3.14159 / 26
400 A3 = A3 / RA
410  DIM N(12)
420  FOR I = 1 TO 12
430  READ N(I)
440  NEXT I
450  DATA    0,31,59,90,120,151,181,212,243,273,304,334
460  REM    EXTRA DAY IN LEAP YEARS IGNORED
480  IF A4 < 0 THEN A5$ = " S LAT;  "
490  IF A4 =  > 0 THEN A5$ = " N LAT;  "
500  IF O3 < 0 THEN O5$ = " E LONG"
510  IF O3 =  > 0 THEN O5$ = " W LONG"
520 A4 =  INT (A4 * 100 + .5) / 100
530 O4 =  INT (O3 * 100 + .5) / 100
540 B$ = "AT THIS LATITUDE THE SUN DOESN'T SET ON: "
550 C$ = "AT THIS LATITUDE THE SUN DOESN'T RISE ON: "
560 F$ = "     DATE        SUNRISE       SUNSET      SUN'S DECL    HRS SUNLIGHT"
570 J$ = "    MO DAY      HRS  MINS     HRS MINS       DEGREES      HRS MINS"
580  IF WD$ = "Y" THEN  GOSUB 1230
590  IF WD$ <  > "Y" THEN  GOSUB 1360
600  IF LP$ = "Y" THEN  PRINT D$;P1$: PRINT I$;X$: GOSUB 1230
610 LN = 19
620 K = N(M) + DA
630  FOR Q = 1 TO NR
640 LN = LN + 1
650  IF LN = 60 AND LP$ = "Y" THEN  GOSUB 1180
660 DT = K - N(M)
670  IF DT = 32 THEN DT = 1
680  IF K < 366 THEN 700
690 M = 1:DA = 1:K = 1
700  IF M = 12 THEN 730
710  IF K > N(M + 1) THEN DT = K - (N(M + 1))
720  IF K > N(M + 1) THEN M = M + 1
730 X = P * K / 7
740 K = K + 1
750 D = .456 - 22.915 *  COS (X) - .43 *  COS (2 * X) - .156 *  COS (3 * X) + 3.83
 *  SIN (X) + .06 *  SIN (2 * X) - .082 *  SIN (3 * X)
755 E = .008 + .51 *  COS (X) - 3.197 *  COS (2 * X) - .106 *  COS (3 * X) - .15 *
  COS (4 * X) - 7.317 *  SIN (X) - 9.471 *  SIN (2 * X) - .391 *  SIN (3 * X) - .2
42 *  SIN (4 * X)
760  REM  D IS DECLINATION OF SUN IN DEGREES; E IS EQUATION OF TIME IN MINUTES
770 DE = D / RA
780 Y =  TAN (DE) *  TAN (A3)
```

```
790  IF Y => 1 THEN 1130
800  IF Y <=  - 1 THEN 1130
810 AR =  -  ATN (Y /  SQR ( - Y * Y + 1)) + 1.5707633
820 AC = AR * RA
830 RI = 03 / 15 + AC / 15:RI = RI - E / 60
840 ST = 03 / 15 - AC / 15:ST = ST - E / 60
850 LT = ST - RI
860  IF LT < 0 THEN LT = 24 + LT
870  IF (LT -  INT (LT)) > .99 THEN LT = ( INT ((LT * 100) + .6)) / 100
880 LH =  INT (LT)
890 LM = (LT -  INT (LT)) * 60
900  IF RI > 24 THEN RI = RI - 24
910  IF RI < 0 THEN RI = RI + 24
920  IF (RI -  INT (RI)) > .99 THEN RI = ( INT ((RI * 100) + .6)) / 100
930 R1 =  INT (RI)
940 R2 = (RI -  INT (RI)) * 60
950  IF ST > 24 THEN ST = ST - 24
960  IF ST < 0 THEN ST = ST + 24
970  IF (ST -  INT (ST)) > .99 THEN ST = ( INT ((ST * 100) + .6)) / 100
980 S1 =  INT (ST)
990 S2 = (ST -  INT (ST)) * 60
1000 R2 =  INT (R2):S2 =  INT (S2):LM =  INT (LM)
1010 D =  INT (D * 100 + .5) / 100
1020 R1$ =  STR$ (R1):R2$ =  STR$ (R2)
1030  IF R1 < 10 THEN R1$ = "0" +  STR$ (R1)
1050  IF R2 < 10 THEN R2$ = "0" +  STR$ (R2)
1060 S1$ =  STR$ (S1):S2$ =  STR$ (S2):LH$ =  STR$ (LH):LM$ =  STR$ (LM)
1070  IF S1 < 10 THEN S1$ = "0" +  STR$ (S1)
1080  IF S2 < 10 THEN S2$ = "0" +  STR$ (S2)
1090  IF LH < 10 THEN LH$ = "0" +  STR$ (LH)
1100  IF LM < 10 THEN LM$ = "0" +  STR$ (LM)
1110 RX$ = R1$ + ":" + R2$:SX$ = S1$ + ":" + S2$:LX$ = LH$ + ":" + LM$
1130  IF LP$ = "Y" THEN  PRINT D$;P1$: GOSUB 1310
1140  GOSUB 1430
1150  NEXT Q
1160  IF LP$ = "Y" THEN  PRINT D$;"PR#1": GOSUB 1470
1170  END
1180  PRINT D$;P1$
1190  FOR J = 1 TO 11: PRINT : NEXT J
1200  PRINT F$: PRINT J$: PRINT :LN = 8
1210  PRINT D$;P$
1220  RETURN
1230  PRINT "    THIS COMPUTATION IS FOR: "; ABS (A4);A5$; ABS (O4);O5$
1250  PRINT : PRINT "    GEOGRAPHIC LOCATION: ";G$
1260  PRINT : PRINT
1270  PRINT  TAB( 4);B3$: PRINT : PRINT
1280  PRINT F$: PRINT J$: PRINT
1290  PRINT D$;P$
1300  RETURN
1310  IF Y > 1 THEN  PRINT I$;X$; TAB( 4);B$;M;"/";DT: GOTO 1340
1320  IF Y <  - 1 THEN  PRINT I$;X$; TAB( 4);C$;M;"/";DT: GOTO 1340
1330  PRINT I$;X$; SPC( 4);M;"/";DT; SPC( 13 - ( LEN ( STR$ (M)) +  LEN ( STR$ (DT
))));RX$; SPC( 8);SX$; SPC( 10);D; SPC( 13 -  LEN ( STR$ (D)));LX$
1340  PRINT D$;P$
1350  RETURN
1360  PRINT : PRINT
1370  PRINT "COORDINATES: "; ABS (A4);A5$; ABS (O4);O5$
1380  PRINT : PRINT "LOCATION: ";G$
1390  PRINT : PRINT B3$: PRINT : PRINT
1400  PRINT "DATE   RISE     SET    INCLIN  HRS SUN"
1410  PRINT
1420  RETURN
1430  IF Y > 1 THEN  PRINT B$;M;"/";DT: GOTO 1460
1440  IF Y <  - 1 THEN  PRINT C$;M;"/";DT: GOTO 1460
1450  PRINT M;"/";DT; TAB( 8);RX$; TAB( 17);SX$; TAB( 25);D; TAB( 33);LX$
1460  RETURN
1470  FOR J = LN TO 65: PRINT : NEXT J
1480  PRINT D$;P$
1490  RETURN
```

THE DX CHECKER

What it does

This short program, which will run on almost any computer, calculates the distance and direction from one point on the globe to any other point. For that matter, it will compute the distance and direction from one point on your block to another—if your coordinates are precise enough. It is ideal for VHF DXers who need accurate distance and heading data over paths of a few dozen or a few hundred miles. It uses the same math as the BEAMHEADING and DX DISPLAY programs, but without the LATLONG data file and printout capabilities.

Setup instructions

For most computers, the program will run as listed with no modifications at all. Timex/Sinclair users must modify the INPUT statements as described in the last chapter so that each one is on a line separate from its prompting question.

Operating notes

As in the other latitude-longitude programs, a minus sign is required for both degrees and minutes to indicate east longitude or south latitude. For accurate close-in calculations, it is especially important to enter precise data. Line 280, which rounds the results to whole numbers, should be eliminated for local work where high precision is desired.

Modifications for CP/M

None—the listed version of this program should run without modification on most microcomputers, including Timex computers with additional RAM. It will run on a VIC-20 without any RAM expansion, provided you modify lines 50 and 60 for separate inputs as explained in Chapter 5.

DX CHECKER (standard version)

```
10 REM DIRECT-ENTRY AZIMUTH/DISTANCE CHECKER
20 PI=3.14159:RAD=57.2958
30 PRINT:PRINT "ENTER YOUR COORDINATES IN DEGREES, MINUTES"
40 PRINT "USE '-' FOR BOTH DEGREES AND MINS"
42 PRINT "IF SOUTH LATITUDE, EAST LONGITUDE"
50 INPUT "LATITUDE:  (E.G. 33,30)";LA,AM
60 INPUT "LONGITUDE:  (E.G.116,30)";LO,OM
70 LT=(LA+AM/60)/RAD:LG=(LO+OM/60)/RAD
80 PRINT:PRINT "NOW ENTER THE OTHER LOCATION'S COORDINATES"
90 INPUT "LATITUDE:  ";L1,M1
100 INPUT "LONGITUDE: ";L2,M2
110 A1=(L1+M1/60)/RAD:O1=(L2+M2/60)/RAD
120 L=LG-O1:IF L=0 THEN 300
130 IF L>PI THEN L=L-2*PI
140 IF L<-PI THEN L=L+2*PI
150 AA=COS(L)*(COS(A1)/SIN(A1)):AA=ATN(AA)
160 AZ={(COS(L)/SIN(L))*COS(LT+AA)}/SIN(AA):AZ=ATN(1/AZ)
170 AZ=AZ*RAD:L=L*RAD
180 IF L>0 AND AZ<0 THEN AZ=AZ+180
190 IF L<0 AND AZ>0 THEN AZ=AZ+180
200 IF L<0 AND AZ<0 THEN AZ=AZ+360
210 AZ=INT(AZ+.5)
220 DX=(SIN(LT)*SIN(A1))+(COS(LT)*COS(A1)*COS(L/RAD))
230 DX=SQR(1-(DX*DX))/DX
240 IF LG=O1 THEN 320
250 DX=ATN(DX)
260 DX=DX*RAD:IF DX<0 AND DX>-90 THEN DX=DX+180
```

```
270 DX=DX*69.0468
280 DX=INT(DX+.5)
290 GOTO 340
300 IF LT>A1 THEN AZ=180:GOTO 210
310 IF LT<A1 THEN AZ=0:GOTO 210
320 IF LT>A1 THEN DX=LT-A1:GOTO 260
330 IF LT<A1 THEN DX=A1-LT:GOTO 260
340 PRINT "TO ";L1;"/";M1;"LAT  AND  ";L2;"/";M2;" LONG"
350 PRINT "DISTANCE=      ";DX;" STAT. MILES"
360 PRINT "BEAM HEADING= ";AZ;" DEGREES"
```

CHAPTER

8

PROGRAMS FOR THE CONTEST OPERATOR

For many radio amateurs the greatest challenge of all is contest operating. In a contest you can see just how good your station really is and match operating skills with the best operators in the country—or the world.

In this competitive environment, winners and losers are often separated by the narrowest of margins. Just one extra multiplier may put the winner ahead of the pack. That's why contest stations, like race cars, must be finely tuned. A contest station may not be the most expensive station in town, but on a given day it's likely to be the first one to get through a pile-up. The ham who makes contesting a specialty wants every bit of efficiency possible from his amplifier and demands the best in sensitivity, selectivity, and overload protection of his receiver. The antennas? Well, they too must be tuned for optimum performance. Some contesters think nothing of installing a new antenna system just for one contest, only to take it down when the contest is over.

But the avid contester's arsenal doesn't include just antennas, transmitters, and receivers. Many now have a new weapon—a personal computer. Computerized on-line logging and duplicate-checking eliminates almost all of the tedious paperwork of contesting. Once and for all it ends the time-consuming post-contest chore of cross-checking your logs for overlooked multipliers and "dupes."

But even more important than its advantages after the contest is the edge a computer can give you during a contest. Either the duplicate-checking or logging programs in this chapter will allow you to instantly

When they log a contest on a computer, many operators use headphones, a boom microphone, and a foot switch for send-receive switching. Seasoned contest operators have done that for years, of course, but it's an especially good technique in computer-assisted contest operating.

see if you've already worked any station you hear during a contest. Your computer will even tell you if you need the station on other bands. The DUPECHECKER program will work with any amateur radio contest.

The logging programs—which are custom tailored to the rules of the most popular contests—allow you to log an entire contest on a computer. Not only can you quickly determine if any call sign is a "dupe," but you can also get an immediate listing of the multipliers you have on a given band—or a display of your running score up to that point. Of course, you can also find out quickly if there are other bands on which you need a particular station or multiplier. And when it's all over, the computer will print out your complete contest log and dupesheets without any hassle—and without illegible erasures. You can show up on a post-contest net knowing your score and multiplier tally, with all dupes removed!

If you're an experienced contest operator, you realize what a significant advantage it is to know your score at any given moment—or to have a running list of multipliers on each band. These programs give the computerized contester a big edge over the guy who's still fumbling with a clumsy map-sized dupesheet and 40 log pages. (Just where *is* page 36? Is that the one I dropped behind the desk???)

The advantages of computerized logging are even greater for a multioperator group. One operator can concentrate entirely on the job of operating at the fastest possible pace, while another mans the computer console for instant logging and dupechecking. And because of the inherent appeal of a computer, the once-onerous chore of being the "second op" becomes fun.

If you're even casually interested in contesting, you'll find that competitive operating is a brand new ball game when you add a personal computer to your team.

THE DUPECHECKER

What it does

The nemesis of the contest operator is the "dupe." You can only work each station once per band in most contests. In some, you're only permitted to work a station once during the entire contest, regardless of band. If the same call shows up in your log a second time, it's a dupe. If your log includes too many dupes, you can be disqualified or penalized. Moreover, dupes make it impossible to keep track of your score. Many a good operator has finished a contest thinking he'd set a new record, only to see his log shrink back to non-record proportions when the dupes were removed.

The DUPECHECKER was designed to solve these problems. Its primary function is to store thousands of call signs in RAM and almost instantly check each new call against the entire list. Even with 1,500 calls in memory, a new call can be checked in less than two seconds on a relatively slow 6502-based computer.

We've seen other dupechecking programs written in BASIC, but they are too slow for serious contest use. The DUPECHECKER avoids this problem by using some innovative programming techniques. First, call signs are converted to integers; they are not stored as strings. This avoids the very slow process of searching and comparing strings. An added benefit of this approach is that it keeps the string pool very small, which means the program is not subject to the lengthy "garbage collections" that can cause a program running in BASIC to seemingly "go dead" for several minutes. With this program, the BASIC garbage collections are so fast that they are hardly ever obvious.

In addition, the hashing algorithm used in DUPECHECKER greatly speeds the search process. As a result, this program will dupe-check 4,000 call signs in less than five seconds on a typical 6502-based computer. On a 4 MHz Z80A machine, it will dupe-check 1,500 calls in less than one second and 4,000 calls in about two seconds. And if the DUPECHECKER is compiled with a BASIC compiler (see Chapter 4), the response is almost instantaneous even with thousands of calls in RAM. On most personal computers, incidentally, this program will store more than 4,000 calls, keeping track of them on a band-by-band basis.

When the contest is over, a companion program (DUPEPRINT) will read the data files created by DUPECHECKER and sort the calls either alphabetically or by call area, producing neat "dupesheets."

Setup Instructions

For a program as sophisticated as this, the DUPECHECKER requires remarkably few modifications in most cases. The Apple, Commodore 64, and TRS-80 listings use virtually no commands or syntax likely to cause problems or require modifications for variations in hardware. If you have any of these computers with 48K or more of RAM and a disk drive, you're all set.

With a few minor modifications described later (most of which involve the video display, not the way the program actually runs), the DUPECHECKER is readily adaptable to the IBM PC and CP/M systems running Microsoft BASIC-80.

As listed, the DUPECHECKER will write its data files on the first disk drive of an Apple, and on the logged drive of other systems. If you want the data files to use a second disk drive, change the value of DV$ in line 90 of the Apple version. In the TRS-80 and CP/M listing, the filenames in lines 1230, 1240, 1320, and 1360 could be modified to include a disk drive designation such as ":1" for TRS-80 or "B:" in a CP/M system.

If you have two or more disk drives, it is preferable to use a second drive to store data files. This allows you to use a completely blank disk, thus maximizing the storage capacity. With the DUPECHECKER, however, disk storage capacity is not a serious problem: a single disk will hold more than the 4,000 calls that most microcomputers can handle in memory when running this program.

Operating notes

In this chapter, we have given separate editorial treatment to the operating instructions for both the DUPECHECKER and the three major contest loggers. The startup instructions include extensive "prompts" to tell the operator what to do, and the main menu summarizes the commands available during an operating session.

The main thing that should be emphasized here is the importance of using a new diskette when running these programs, not one cluttered with other files. This program and especially the loggers generate large data files.

Modifications for CP/M

For CP/M systems running Microsoft BASIC-80, the TRS-80 and CP/M listing of the DUPECHECKER should be modified as follows:

1. Delete the clear statement in line 35;
2. Add your system's video highlighting and bell command codes to line 50 (HI$ is for inverse; LO$ is for normal video):
3. Replace the clear-screen command ("CLS") in lines 120, 1590 and 1670 with the proper command for your system;
4. Modify the formatting for a more attractive display as outlined in the next paragraph.

The TRS-80 graphic character set and display commands will need to be modified for most CP/M systems. The main title that appears at sign-on uses the TRS-80's double-size type display. Since that will not work on other systems, lines 1600–1630 should be revised to center the heading, and preferably make it display in inverse video.

Also, the main menu and the "IT'S-A-DUPE" warning use graphic characters for highlighting. In lines 1700, 2010, and 2030, ASCII codes 131 and 140 (following STRING$) should be replaced with 42, which will produce a row of "****". And finally, line 1840 (which prevents the menu from scrolling off the 16-line TRS-80 screen) can be eliminated.

DUPECHECKER (TRS-80 and CP/M)

```
10 REM THE CONTEST DUPECHECKER (TRS-80 VERSION)
20 REM BY KC6A AND N6NB
30 REM COPYRIGHT 1983 BY HAYDEN BOOK COMPANY, INC.
35 CLEAR 200
40 AA=ASC("A")-1:A0=ASC("A")-ASC("0")+26
50 BL$="":HI$="":LO$="":REM NEEDED; DON'T DELETE
90 REM DATA FILES WILL BE WRITTEN ON DEFAULT DRIVE
```

```
100 GOSUB 1580
110 A$="":PRINT:INPUT "NEW CONTEST (N) OR RESUME (R) ";A$
120 CLS:PRINT
130 IF A$="N" THEN 180
140 IF A$<>"R" THEN PRINT BL$:GOTO 100
150 GOSUB 1320
160 IF HC$="Y" THEN GOSUB 1430
170 SS=26*NL+1:GOTO 390
180 HC$="":INPUT "WANT A HARD COPY WHILE LOGGING (Y/N) ";HC$
190 IF HC$="Y" THEN INPUT "TURN ON PRINTER. <CR> WHEN READY ";A$:GOSUB 1990
200 T=1:PRINT
210 INPUT "WHAT IS THE CONTEST NAME ";Y$
220 PRINT:Z$="":INPUT "WHAT IS THE DATE (MM/DD/YY) ";Z$:PRINT
230 IF HC$="Y" THEN GOSUB 1430
240 INPUT "HOW MANY SEPARATE DUPE SHEETS DO YOU NEED ";NL:PRINT
250 IF NL<1 THEN PRINT BL$:GOTO 240
260 DIM LN$(NL)
270 M1=INT((FRE(0)-2000)/6)
280 IF NL*26>=M1 THEN PRINT "TOO MANY";BL$:GOTO 240
290 DIM C1%(M1),C2%(M1),P%(M1)
300 IF NL=1 THEN LN$(1)="1":GOTO 360
310 FOR J=1 TO NL
320 PRINT "WHAT BAND IS DUPE SHEET #";J
330 INPUT "(E.G. 20, 20CW, OR 20PHONE) ";LN$(J)
340 IF LEN(LN$(J))>20 THEN PRINT BL$;"NAME IS TOO LONG":GOTO 320
350 PRINT:NEXT J
360 SS=26*NL+1:L%=SS-1
370 PRINT:PRINT "SPACE IS AVAILABLE FOR APPROXIMATELY";M1;"CALL SIGNS. GOOD LUCK
    !"
380 FOR K=1 TO L%:C1%(K)=-32000:C2%(K)=-32000:P%(K)=0:NEXT K
390 LG=1:LH%=0:IF NL>1 THEN GOSUB 1050
400 GOSUB 1670
410 PRINT:PRINT "DUPECHECKER READY FOR QSO #";T;"-";LN$(LG)
420 GOSUB 1870:IF X1%=1 THEN 510
430 IF G$="X" THEN 1130
440 IF G$="DEL" THEN 1180
450 IF G$="QRT" THEN GOSUB 1230:END
460 IF G$="QSY" THEN GOSUB 1030:GOTO 410
470 IF G$="SAVE" THEN GOSUB 1230:GOTO 410
480 IF G$="?" THEN 400
490 GOTO 600
500 PRINT "INVALID CALL";BL$:GOTO 410
510 PRINT " ..ABORTED..":GOTO 410
520 PRINT "DUPE SHEET FULL";BL$:GOTO 410
530 PRINT "CAN'T DELETE";BL$:GOTO 410
540 IF DD=1 THEN 1200
550 GOSUB 2010:GOTO 410
600 DD=0
610 E(5)=0:E(6)=0:K=1:IF LEN(G$)>10 THEN 500
620 IF RIGHT$(G$,1)<>"?" THEN 640
630 DD=2:G$=LEFT$(G$,LEN(G$)-1)
640 IF LEN(G$)<4 THEN 500
650 IF MID$(G$,3,1)<"A" THEN 680
660 IF MID$(G$,2,1)=>"A" THEN 500
670 G$=" "+G$:K=2:E(1)=0
680 FOR J=K TO LEN(G$)
690 W$=MID$(G$,J,1)
700 IF W$="/" THEN 750
710 E(J)=ASC(W$)-AA:IF E(J)>0 THEN IF E(J)>26 THEN 500
720 IF W$<"0" THEN 500
730 IF E(J)<1 THEN E(J)=E(J)+A0:IF E(J)>36 THEN 500
740 NEXT J
750 IF J>7 THEN 500
760 IF J-K<4 THEN 500
770 FOR J=4 TO 6:IF E(J)>26 THEN 500
780 NEXT J
790 E1%=-32000+E(1)*1369+E(2)*37+E(3)
800 E2%=-32000+E(4)*1369+E(5)*37+E(6)
810 IF DD=2 GOTO 1470
820 H%=E(4)+LG*26-26
```

```
830 IF C1%(H%)<>-32000 THEN 890
840 IF DD=1 THEN PRINT BL$;"CALL NOT IN DUPE SHEET":GOTO 410
850 GOSUB 970:IF A$<>"" THEN 510
860 LH%=H%
870 C1%(H%)=E1%:C2%(H%)=E2%:P%(H%)=0
880 T=T+1:GOTO 410
890 IF C2%(H%)<>E2% THEN 910
900 IF C1%(H%)=E1% THEN 540:REM DUPE
910 IF P%(H%)=0 THEN 930
920 H%=P%(H%):GOTO 890
930 IF DD=1 THEN 840
940 IF L%=M1 THEN 520
950 GOSUB 970:IF A$="" THEN LH%=H%:L%=L%+1:P%(H%)=L%:H%=L%:GOTO 870
960 GOTO 510
970 PRINT "---CALL OK---   QSO ENTERED"
980 A$="":PRINT "IF NO QSO, TYPE 'X' BELOW"
1000 IF HC$<>"Y" THEN 1020
1010 LPRINT T;"   ";G$
1020 RETURN
1030 LH%=0:IF NL=1 THEN RETURN
1040 GOSUB 1900:IF G$<>"" THEN 1060
1050 INPUT "WHICH BAND DO YOU WANT ";G$
1060 FOR K=1 TO NL:IF G$=LN$(K) THEN 1110
1070 NEXT K
1080 PRINT "NO SUCH DUPE SHEET. VALID SHEETS ARE:";BL$
1090 FOR K=1 TO NL:PRINT TAB(5)LN$(K):NEXT K
1100 PRINT:GOTO 1050
1110 IF HC$="Y" THEN GOSUB 1460
1120 LG=K:RETURN
1130 IF LH%=0 THEN 530
1140 IF LH%<SS THEN IF P%(LH%)=0 THEN C1%(LH%)=-32000:GOTO 1160
1150 P%(LH%)=0:L%=L%-1
1160 IF HC$="Y" THEN LPRINT " *** PREVIOUS CALL DELETED ***"
1170 LH%=0:T=T-1:PRINT "   DELETED":GOTO 410
1180 GOSUB 1900: IF G$="" THEN INPUT "WHAT IS THE CALL TO BE DELETED ";G$
1190 DD=1:GOTO 610
1200 C1%(H%)=-32001:C2%(H%)=-32000
1210 IF HC$="Y" THEN LPRINT "   *** DELETING ";G$
1220 PRINT "   DELETED":GOTO 410
1230 OPEN "O",1,"DUPELOGA"
1240 OPEN "R",2,"DUPELOGB",6
1250 FIELD 2,2 AS C1$,2 AS C2$,2 AS P$
1260 PRINT #1,HC$:PRINT#1,T:PRINT#1,Y$:PRINT#1,Z$:PRINT#1,NL:PRINT#1,L%:PRINT#1,
     M1
1270 FOR J=1 TO NL:PRINT#1,LN$(J):NEXT J
1280 FOR J=1 TO L%
1290 LSET C1$=MKI$(C1%(J)):LSET C2$=MKI$(C2%(J)):LSET P$=MKI$(P%(J))
1300 PUT 2,J:NEXT J
1310 CLOSE:RETURN
1320 OPEN "I",1,"DUPELOGA"
1330 INPUT #1,HC$:INPUT#1,T:INPUT#1,Y$:INPUT#1,Z$:INPUT#1,NL:INPUT#1,L%:INPUT#1,
     M1
1340 DIM LN$(NL),C1%(M1),C2%(M1),P%(M1)
1350 FOR J=1 TO NL:INPUT#1,LN$(J):NEXT J
1360 OPEN "R",2,"DUPELOGB",6
1370 FIELD 2,2 AS C1$,2 AS C2$,2 AS P$
1380 FOR J=1 TO L%
1390 GET 2,J
1400 C1%(J)=CVI(C1$):C2%(J)=CVI(C2$):P%(J)=CVI(P$)
1410 NEXT J
1420 CLOSE:RETURN
1430 LPRINT:LPRINT:LPRINT TAB(5)"CONTEST: ";Y$
1440 LPRINT TAB(5)"DATE:    ";Z$
1450 RETURN
1460 LPRINT:LPRINT "     QSY TO: ";LN$(K):GOTO 1450
1470 GOSUB 2100
1472 PRINT:PRINT "NEEDED ON:"
1480 FOR J=1 TO NL
1490 H%=E(4)+J*26-26
1500 IF C1%(H%)=-32000 THEN 1550
```

```
1510 IF C2%(H%)<>E2% THEN 1530
1520 IF C1%(H%)=E1% THEN 1570
1530 IF P%(H%)=0 THEN 1550
1540 H%=P%(H%):GOTO 1510
1550 PRINT "                    ";
1560 PRINT LN$(J)
1570 NEXT J:PRINT:GOTO 410
1580 FOR J=1 TO 40:PRINT:NEXT J
1590 CLS:PRINT:PRINT
1600 PRINT CHR$(23)
1610 PRINT TAB(8)"THE DUPECHECKER"
1620 PRINT
1630 PRINT "(C) 1983 BY HAYDEN BOOK CO.,INC."
1640 PRINT:PRINT:PRINT:PRINT:PRINT
1660 RETURN
1670 CLS:PRINT:PRINT                                    "
1680 PRINT HI$;"              INSTRUCTIONS                "
1690 PRINT "
1700 PRINT TAB(13)STRING$(12,131)                         "
1710 PRINT "                                              "
1720 PRINT "  -ENTER CALL TO ADD IT TO DUPESHEET-         "
1730 PRINT "                                              "
1740 PRINT " 'N1NNN?' ENTER ANY CALL PLUS '?' FOR         "
1750 PRINT "          BAND-BY-BAND DUPE CHECK             "
1780 PRINT " 'X'    DELETES THE LAST CALL ENTERED         "
1790 PRINT " 'DEL'  DELETES ANY CALL                      "
1800 PRINT " 'QRT'  SAVES ALL CALLS AND EXITS             "
1810 PRINT " 'QSY'  CHANGES BANDS/DUPE SHEETS             "
1820 PRINT " 'SAVE' SAVES CALLS WITHOUT EXITING           "
1830 PRINT " '?'    DISPLAYS THIS MENU                    "
1840 CR$="":INPUT "ENTER <CR> TO CONTINUE";CR$
1850 PRINT LO$
1860 RETURN
1870 GS$="":LINE INPUT "?";GS$:LL=LEN(GS$)
1880 X1%=0:IF INSTR(GS$,"-")>0 THEN X1%=1:RETURN
1890 LP=1
1900 IF LP>LL THEN 1930:REM 2ND ENTRY
1910 FOR LT=LP TO LL:IF MID$(GS$,LT,1)<>" " THEN 1940
1920 NEXT LT
1930 G$="":RETURN
1940 LP=LT
1950 FOR LT=LP TO LL:IF MID$(GS$,LT,1)=" " THEN 1980
1960 NEXT LT
1970 LT=LT+1
1980 G$=MID$(GS$,LP,LT-LP):LP=LT:RETURN
1990 REM PRINTER INIT, IF NEEDED
2000 RETURN
2010 PRINT HI$;STRING$(36,140)
2020 PRINT "** IT'S A DUPE  **  BOO  **  HISS **"
2030 PRINT STRING$(36,131):PRINT LO$
2040 RETURN
2100 H%=E(4)+LG*26-26
2110 IF C1%(H%)=-32000 THEN RETURN
2120 IF C2%(H%)<>E2% THEN 2140
2130 IF C1%(H%)=E1% THEN GOSUB 2010:GOTO 1472
2140 IF P%(H%)=0 THEN RETURN
2150 H%=P%(H%):GOTO 2120
```

DUPECHECKER (Apple)

```
10  REM  THE CONTEST DUPECHECKER (APPLE)
20  REM  BY KC6A AND N6NB
30  REM  (C) 1983 BY HAYDEN BOOK CO.,INC.
40 AA =  ASC ("A") - 1:A0 =  ASC ("A") -  ASC ("0") + 26
50 D$ =  CHR$ (4):BL$ =  CHR$ (7)
60 LP$ = D$ + "PR#1":SCN$ = D$ + "PR#0": REM  PRINTER & SCREEN SELECT
70 RD$ = D$ + "READ ":WR$ = D$ + "WRITE "
```

```
72 OP$ = D$ + "OPEN ":CL$ = D$ + "CLOSE ":DL$ = D$ + "DELETE "
80 F1$ = "DUPELOGA":F2$ = "DUPELOGB":R2$ = ",L19"
90 DV$ = ",D1": REM  DRIVE FOR DATA FILES
100  GOSUB 1580
110  PRINT : INPUT "NEW CONTEST (N) OR RESUME (R)? ";A$
120  PRINT
130  IF A$ = "N" THEN 180
140  IF A$ <  > "R" THEN  PRINT BL$: GOTO 100
150  GOSUB 1320
160  IF HC$ = "Y" THEN  GOSUB 1430
170 SS = 26 * NL + 1: GOTO 390
180  INPUT "WANT A HARD COPY WHILE LOGGING (Y/N)? ";HC$
190  IF HC$ = "Y" THEN  INPUT "TURN ON PRINTER. <CR> WHEN READY ";A$: GOSUB 1990
200 T = 1: PRINT
210  INPUT "WHAT IS THE CONTEST NAME? ";Y$
220  PRINT : INPUT "WHAT IS THE DATE (MM/DD/YY)? ";Z$: PRINT
230  IF HC$ = "Y" THEN  GOSUB 1430
240  PRINT "HOW MANY SEPARATE DUPE"
242  INPUT "SHEETS DO YOU NEED? ";NL: PRINT
250  IF NL < 1 THEN  PRINT BL$: GOTO 240
260  DIM LN$(NL)
262 M1 =  FRE (0): IF M1 < 0 THEN M1 = M1 + 65536
270 M1 =  INT ((M1 - 2000) / 6)
280  IF NL * 26 >  = M1 THEN  PRINT "TOO MANY";BL$: END
290  DIM C1%(M1),C2%(M1),P%(M1)
300  IF NL = 1 THEN LN$(1) = "1": GOTO 360
310  FOR J = 1 TO NL
320  PRINT "WHAT BAND IS DUPE SHEET #";J
330  INPUT "(E.G. 20, 20CW, OR 20PHONE) ";LN$(J)
340  IF  LEN (LN$(J)) > 20 THEN  PRINT BL$;"NAME IS TOO LONG": GOTO 320
350  PRINT : NEXT J
360 SS = 26 * NL + 1:L% = SS - 1
370  PRINT : PRINT "SPACE IS AVAILABLE FOR APPROXIMATELY": PRINT M1;" CALL SIGNS.
GOOD LUCK!"
380  FOR K = 1 TO L%:C1%(K) =  - 32000:C2%(K) =  - 32000:P%(K) = 0: NEXT K
390 LG = 1:LH% = 0: IF NL > 1 THEN  PRINT : GOSUB 1050
400  GOSUB 1670
410  PRINT : PRINT "DUPECHECKER READY FOR QSO#";T;" - ";LN$(LG)
420  GOSUB 1870: IF X1% = 1 THEN 510
430  IF G$ = "X" THEN 1130
440  IF G$ = "DEL" THEN 1180
450  IF G$ = "QRT" THEN  GOSUB 1230: END
460  IF G$ = "QSY" THEN  GOSUB 1030: GOTO 410
470  IF G$ = "SAVE" THEN  GOSUB 1230: GOTO 410
480  IF G$ = "?" THEN 400
490  GOTO 600
500  PRINT "INVALID CALL";BL$: GOTO 410
510  PRINT " ..ABORTED..": GOTO 410
520  PRINT "DUPE SHEET FULL";BL$: GOTO 410
530  PRINT "CAN'T DELETE";BL$: GOTO 410
540  IF DD = 1 THEN 1200
550  GOSUB 2010: GOTO 410
600 DD = 0
610 E(5) = 0:E(6) = 0:K = 1: IF  LEN (G$) > 10 THEN 500
620  IF  RIGHT$ (G$,1) <  > "?" THEN 640
630 DD = 2:G$ =  LEFT$ (G$, LEN (G$) - 1)
640  IF  LEN (G$) < 4 THEN 500
650  IF  MID$ (G$,3,1) < "A" THEN 680
660  IF  MID$ (G$,2,1) =  > "A" THEN 500
670 G$ = " " + G$:K = 2:E(1) = 0
680  FOR J = K TO  LEN (G$)
690 W$ =  MID$ (G$,J,1)
700  IF W$ = "/" THEN 750
710 E(J) =  ASC (W$) - AA: IF E(J) > 0 THEN  IF E(J) > 26 THEN 500
720  IF W$ < "0" THEN 500
730  IF E(J) < 1 THEN E(J) = E(J) + A0: IF E(J) > 36 THEN 500
740  NEXT J
750  IF J > 7 THEN 500
760  IF J - K < 4 THEN 500
770  FOR J = 4 TO 6: IF E(J) > 26 THEN 500
780  NEXT J
```

```
790 El% =  - 32000 + E(1) * 1369 + E(2) * 37 + E(3)
800 E2% =  - 32000 + E(4) * 1369 + E(5) * 37 + E(6)
810  IF DD = 2 THEN 1470
820 H% = E(4) + LG * 26 - 26
830  IF C1%(H%) <  > - 32000 THEN 890
840  IF DD = 1 THEN  PRINT BL$;"CALL NOT IN DUPE SHEET": GOTO 410
850  GOSUB 970: IF A$ <  > "" THEN 510
860 LH% = H%
870 C1%(H%) = El%:C2%(H%) = E2%:P%(H%) = 0
880 T = T + 1: GOTO 410
890  IF C2%(H%) <  > E2% THEN 910
900  IF C1%(H%) = El% THEN 540: REM  DUPE
910  IF P%(H%) = 0 THEN 930
920 H% = P%(H%): GOTO 890
930  IF DD = 1 THEN 840
940  IF L% = M1 THEN 520
950  GOSUB 970: IF A$ = "" THEN LH% = H%:L% = L% + 1:P%(H%) = L%:H% = L%: GOTO 870
960  GOTO 510
970  PRINT "--CALL OK--  QSO ENTRERED"
980 A$ = "": PRINT "IF NO QSO, TYPE 'X' BELOW"
1000  IF HC$ <  > "Y" THEN 1020
1010  PRINT LP$: PRINT T;"   ";G$: PRINT SCN$
1020  RETURN
1030 LH% = 0: IF NL = 1 THEN  RETURN
1040  GOSUB 1900: IF G$ <  > "" THEN 1060
1050  INPUT "WHICH BAND DO YOU WANT? ";G$
1060  FOR K = 1 TO NL: IF G$ = LN$(K) THEN 1110
1070  NEXT K
1080  PRINT "NO SUCH DUPE SHEET. VALID SHEETS ARE:";BL$
1090  FOR K = 1 TO NL: PRINT  TAB( 5)LN$(K): NEXT K
1100  PRINT : GOTO 1050
1110  IF HC$ = "Y" THEN  GOSUB 1460
1120 LG = K: RETURN
1130  IF LH% = 0 THEN 530
1140  IF LH% < SS THEN  IF P%(LH%) = 0 THEN C1%(LH%) =  - 32000: GOTO 1160
1150 P%(LH%) = 0:L% = L% - 1
1160  IF HC$ = "Y" THEN  PRINT LP$: PRINT " *** PREVIOUS CALL DELETED ***": PRINT
SCN$
1170 LH% = 0:T = T - 1: PRINT "   DELETED": GOTO 410
1180  GOSUB 1900: IF G$ = "" THEN  INPUT "WHAT IS THE CALL TO BE DELETED? ";G$
1190 DD = 1: GOTO 610
1200 C1%(H%) =  - 32001:C2%(H%) =  - 32000
1210  IF HC$ = "Y" THEN  PRINT LP$: PRINT "   *** DELETING ";G$: PRINT SCN$
1220  PRINT "   DELETED": GOTO 410
1230  PRINT OP$;F1$;DV$
1232  PRINT DL$;F1$
1234  PRINT OP$;F1$
1236  PRINT WR$;F1$
1260  PRINT HC$: PRINT T: PRINT Y$: PRINT Z$: PRINT NL: PRINT L%: PRINT M1
1270  FOR J = 1 TO NL: PRINT LN$(J): NEXT J
1272  PRINT CL$
1274  PRINT OP$;F2$;DV$
1276  PRINT DL$;F2$
1278  PRINT OP$;F2$;R2$;DV$
1280  FOR J = 1 TO L%
1290  PRINT WR$;F2$;",R";J
1300  PRINT C1%(J): PRINT C2%(J): PRINT P%(J)
1305  NEXT J
1310  PRINT CL$: RETURN
1320  PRINT OP$;F1$;DV$
1322  PRINT RD$;F1$
1330  INPUT HC$: INPUT T: INPUT Y$: INPUT Z$: INPUT NL: INPUT L%: INPUT M1
1340  DIM LN$(NL),C1%(M1),C2%(M1),P%(M1)
1350  FOR J = 1 TO NL: INPUT LN$(J): NEXT J
1360  PRINT CL$
1362  PRINT OP$;F2$;R2$;DV$
1380  FOR J = 1 TO L%
1390  PRINT RD$;F2$;",R";J
1400  INPUT C1%(J): INPUT C2%(J): INPUT P%(J)
1410  NEXT J
1420  PRINT CL$: RETURN
```

```
1430  PRINT LP$: PRINT : PRINT  SPC( 4)"CONTEST: ";Y$
1440  PRINT  SPC( 4)"DATE:     ";Z$
1450  PRINT SCN$: RETURN
1460  PRINT LP$: PRINT : PRINT "      QSY TO: ";LN$(K): GOTO 1450
1470  GOSUB 2100
1472  PRINT : PRINT "NEEDED ON:"
1480  FOR J = 1 TO NL
1490 H% = E(4) + J * 26 - 26
1500  IF C1%(H%) =  - 32000 THEN 1550
1510  IF C2%(H%) <  > E2% THEN 1530
1520  IF C1%(H%) = E1% THEN 1570
1530  IF P%(H%) = 0 THEN 1550
1540 H% = P%(H%): GOTO 1510
1550  PRINT  SPC( 12);
1552  IF J = LG THEN  INVERSE
1560  PRINT LN$(J)
1570  NORMAL : NEXT J: PRINT : GOTO 410
1580  HOME
1590  PRINT : INVERSE
1600  GOSUB 1860
1610  PRINT "           THE DUPECHECKER              "
1620  GOSUB 1860
1630  PRINT "   (C) 1983 BY HAYDEN BOOK CO.,INC.    "
1640  GOSUB 1860
1660  NORMAL : RETURN
1670  HOME : PRINT : INVERSE
1680  GOSUB 1860
1690  PRINT "              INSTRUCTIONS              "
1700  GOSUB 1860: GOSUB 1860
1720  PRINT "  -ENTER CALL TO ADD IT TO DUPESHEET-   "
1730  GOSUB 1860
1740  PRINT " 'N1NNN?' -ENTER ANY CALL PLUS '?' FOR "
1750  PRINT "          BAND-BY-BAND DUPECHECK-       "
1760  GOSUB 1860
1780  PRINT " 'X'    DELETES THE LAST CALL ENTERED  "
1790  PRINT " 'DEL'  DELETES ANY CALL               "
1800  PRINT " 'QRT'  SAVES ALL CALLS AND EXITS      "
1810  PRINT " 'QSY'  CHANGES BANDS/DUPE SHEETS      "
1820  PRINT " 'SAVE' SAVES CALLS WITHOUT EXITING    "
1830  PRINT " '?'    DISPLAYS THIS MENU             "
1840  GOSUB 1860
1850  NORMAL : RETURN
1860  PRINT "                                      ": RETURN
1870  INPUT "?";GS$:LL =  LEN (GS$): IF LL = 0 THEN 1930
1880 X1% = 0: FOR J = 1 TO LL: IF  MID$ (GS$,J,1) = "-" THEN 510
1882  NEXT J
1890 LP = 1
1900  IF LP > LL THEN 1930: REM  2ND ENTRY
1910  FOR LT = LP TO LL: IF  MID$ (GS$,LT,1) <  > " " THEN 1940
1920  NEXT LT
1930 G$ = "": RETURN
1940 LP = LT
1950  FOR LT = LP TO LL: IF  MID$ (GS$,LT,1) = " " THEN 1980
1960  NEXT LT
1970 LT = LT + 1
1980 G$ =  MID$ (GS$,LP,LT - LP):LP = LT: RETURN
1990  REM  PRINTER INIT, IF NEEDED
2000  RETURN
2010  FLASH
2012  PRINT "************************************"
2020  PRINT "** IT'S A DUPE  **  BOO  **  HISS **"
2030  PRINT "************************************"
2040  NORMAL : PRINT BL$: RETURN
2100 H% = E(4) + LG * 26 - 26
2110  IF C1%(H%) =  - 32000 THEN  RETURN
2120  IF C2%(H%) <  > E2% THEN 2140
2130  IF C1%(H%) = E1% THEN  GOSUB 2010: RETURN
2140  IF P%(H%) = 0 THEN  RETURN
2150 H% = P%(H%): GOTO 2120
```

DUPECHECKER (Commodore 64)

```
10 REM THE DUPECHECKER (C64)
20 REM BY KC6A AND N6NB
30 REM COPYRIGHT 1983 BY HAYDEN BOOK COMPANY, INC.
32 OPEN 130,2,2,CHR$(136)
33 OPEN15,8,15
40 AA=ASC("A")-1:A0=ASC("A")-ASC("0")+26
50 BL$=CHR$(7):REM BELL
60 HI$=CHR$(18):REM PUT YOUR VIDEO HIGHLIGHTING COMMAND HERE
70 LO$=CHR$(146):REM PUT YOUR VIDEO NORMAL COMMAND HERE
80 C7$=CHR$(147):REM CLEAR SCREEN
100 GOSUB 1580
110 PRINT:INPUT "NEW CONTEST (N) OR RESUME (R) ";A$
120 PRINT
130 IF A$="N" THEN 180
140 IF A$<>"R" THEN PRINT BL$:GOTO 100
150 GOSUB 1320
160 IF HC$="Y" THEN GOSUB 1430
170 SS=26*NL+1:GOTO 390
180 PRINT#15,"S0:DUPELOGA":PRINT#15,"S0:DUPELOGB"
185 INPUT "WANT A HARD COPY WHILE LOGGING (Y/N) ";HC$
190 IF HC$="Y" THEN INPUT "TURN ON PRINTER. <CR> WHEN READY ";A$:GOSUB 1990
200 T=1:PRINT
210 INPUT "WHAT IS THE CONTEST NAME ";Y$
220 PRINT:INPUT "WHAT IS THE DATE (MM/DD/YY) ";Z$:PRINT
230 IF HC$="Y" THEN GOSUB 1430
240 INPUT "HOW MANY SEPARATE DUPE SHEETS";NL
242 PRINT
250 IF NL<1 THEN PRINT BL$:GOTO 240
260 DIM LN$(NL)
270 M1=INT((FRE(0)-2000)/6)
280 IF NL*26>=M1 THEN PRINT "TOO MANY";BL$:GOTO 240
290 DIM C1%(M1),C2%(M1),P%(M1)
300 IF NL=1 THEN LN$(1)="1":GOTO 360
310 FOR J=1 TO NL
320 PRINT "WHAT BAND IS DUPE SHEET #";J
330 INPUT "(E.G. 20, 20CW, OR 20PHONE) ";LN$(J)
340 IF LEN(LN$(J))>20 THEN PRINT BL$;"NAME IS TOO LONG":GOTO 320
350 PRINT:NEXT J
360 SS=26*NL+1:L%=SS-1
370 PRINT:PRINT "SPACE IS AVAILABLE FOR APPROXIMATELY";M1;"CALL SIGNS. GOOD LUCK!"
380 FOR K=1 TO L%:C1%(K)=-32000:C2%(K)=-32000:P%(K)=0:NEXT K
390 LG=1:LH%=0:IF NL>1 THEN GOSUB 1050
400 GOSUB 1670
410 PRINT:PRINT "SUPER DUPER READY FOR QSO #";T;"-";LN$(LG)
420 GOSUB 1870:IF X1%=1 THEN 510
430 IF G$="X" THEN 1130
440 IF G$="DEL" THEN 1180
450 IF G$="QRT" THEN GOSUB 1230:CLOSE 130:GOSUB6000:CLOSE15:END
460 IF G$="QSY" THEN GOSUB 1030:GOTO 410
470 IF G$="SAVE" THEN GOSUB 1230:GOTO 410
480 IF G$="?" THEN 400
490 GOTO 600
500 PRINT "INVALID CALL";BL$:GOTO 410
510 PRINT " ..ABORTED..":GOTO 410
520 PRINT "DUPE SHEET FULL";BL$:GOTO 410
530 PRINT "CAN'T DELETE";BL$:GOTO 410
540 IF DD=1 THEN 1200
550 GOSUB 2010:GOTO 410
600 DD=0
610 E(5)=0:E(6)=0:K=1:IF LEN(G$)>10 THEN 500
620 IF RIGHT$(G$,1)<>"?" THEN 640
630 DD=2:G$=LEFT$(G$,LEN(G$)-1)
640 IF LEN(G$)<4 THEN 500
650 IF MID$(G$,3,1)<"A" THEN 680
660 IF MID$(G$,2,1)=>"A" THEN 500
670 G$=" "+G$:K=2:E(1)=0
680 FOR J=K TO LEN(G$)
```

```
690 W$=MID$(G$,J,1)
700 IF W$="/" THEN 750
710 E(J)=ASC(W$)-AA:IF E(J)>0 THEN IF E(J)>26 THEN 500
720 IF W$<"0" THEN 500
730 IF E(J)<1 THEN E(J)=E(J)+A0:IF E(J)>36 THEN 500
740 NEXT J
750 IF J>7 THEN 500
760 IF J-K<4 THEN 500
770 FOR J=4 TO 6:IF E(J)>26 THEN 500
780 NEXT J
790 E1%=-32000+E(1)*1369+E(2)*37+E(3)
800 E2%=-32000+E(4)*1369+E(5)*37+E(6)
810 IF DD=2 GOTO 1470
820 H%=E(4)+LG*26-26
830 IF C1%(H%)<>-32000 THEN 890
840 IF DD=1 THEN PRINT BL$;"CALL NOT IN DUPE SHEET":GOTO 410
850 GOSUB 970:IF A$<>"" THEN 510
860 LH%=H%
870 C1%(H%)=E1%:C2%(H%)=E2%:P%(H%)=0
880 T=T+1:GOTO 410
890 IF C2%(H%)<>E2% THEN 910
900 IF C1%(H%)=E1% THEN 540:REM DUPE
910 IF P%(H%)=0 THEN 930
920 H%=P%(H%):GOTO 890
930 IF DD=1 THEN 840
940 IF L%=M1 THEN 520
950 GOSUB 970:IF A$="" THEN LH%=H%:L%=L%+1:P%(H%)=L%:H%=L%:GOTO 870
960 GOTO 510
970 PRINT "--CALL OK--  QSO ENTERED."
980 A$="":PRINT "IF NO QSO, TYPE 'X' BELOW"
1000 IF HC$<>"Y" THEN 1020
1010 PRINT#130,T;"  ";G$
1020 RETURN
1030 LH%=0:IF NL=1 THEN RETURN
1040 GOSUB 1900:IF G$<>"" THEN 1060
1050 INPUT "WHICH BAND DO YOU WANT ";G$
1060 FOR K=1 TO NL:IF G$=LN$(K) THEN 1110
1070 NEXT K
1080 PRINT "NO SUCH DUPE SHEET. VALID SHEETS ARE:";BL$
1090 FOR K=1 TO NL:PRINT TAB(5)LN$(K):NEXT K
1100 PRINT:GOTO 1050
1110 IF HC$="Y" THEN GOSUB 1460
1120 LG=K:RETURN
1130 IF LH%=0 THEN 530
1140 IF LH%<SS THEN C1%(LH%)=-32000:GOTO 1160
1150 P%(LH%)=0:L%=L%-1
1160 IF HC$="Y" THEN PRINT#130," *** PREVIOUS CALL DELETED ***"
1170 LH%=0:T=T-1:PRINT "   DELETED":GOTO 410
1180 GOSUB 1900: IF G$="" THEN INPUT "WHAT IS THE CALL TO BE DELETED ";G$
1190 DD=1:GOTO 610
1200 C1%(H%)=-32001:C2%(H%)=-32000
1210 IF HC$="Y" THEN PRINT#130,"   *** DELETING ";G$
1220 PRINT "   DELETED":GOTO 410
1230 OPEN4,8,4, "0:DUPELOGA,S,W":GOSUB6000
1260 PRINT#4,HC$:PRINT#4,T:PRINT#4,Y$:PRINT#4,Z$
1262 PRINT#4,NL:PRINT#4,L%:PRINT#4,M1
1270 FOR J=1 TO NL:PRINT#4,LN$(J):NEXT J
1272 CLOSE4
1274 OPEN3,8,3,"0:DUPELOGB,L,"+CHR$(23):GOSUB6000
1280 FOR J=1 TO L%
1290 RN=J:GOSUB7000
1292 RP=1:GOSUB7100:PRINT#3,C1%(J)
1294 RP=9:GOSUB7100:PRINT#3,C2%(J)
1296 RP=17:GOSUB7100:PRINT#3,P%(J)
1300 NEXT J
1310 CLOSE3:RETURN
1320 OPEN4,8,4,"0:DUPELOGA,S,R":GOSUB6000
1330 INPUT#4,HC$:INPUT#4,T:INPUT#4,Y$:INPUT#4,Z$
1332 INPUT#4,NL:INPUT#4,L%:INPUT#4,M1
1340 DIM LN$(NL),C1%(M1),C2%(M1),P%(M1)
```

```
1350 FOR J=1 TO NL:INPUT#4,LN$(J):NEXT J
1352 GOSUB 6000
1354 CLOSE4
1360 OPEN3,8,3,"0:DUPELOGB,L,"+CHR$(23):GOSUB6000
1380 FOR J=1 TO L%
1390 RN=J:GOSUB7000
1392 RP=1:GOSUB7100:INPUT#3,C1%(J)
1394 RP=9:GOSUB7100:INPUT#3,C2%(J)
1396 RP=17:GOSUB7100:INPUT#3,P%(J)
1410 NEXT J
1420 CLOSE3:RETURN
1430 PRINT#130,"":PRINT#130," CONTEST: ";Y$
1440 PRINT#130,"      DATE:    ";Z$
1450 RETURN
1460 PRINT#130:PRINT#130,"     QSY TO: ";LN$(K):GOTO 1450
1470 GOSUB 2100
1472 PRINT:PRINT "NEEDED ON:"
1480 FOR J=1 TO NL
1490 H%=E(4)+J*26-26
1500 IF C1%(H%)=-32000 THEN 1550
1510 IF C2%(H%)<>E2% THEN 1530
1520 IF C1%(H%)=E1% THEN 1570
1530 IF P%(H%)=0 THEN 1550
1540 H%=P%(H%):GOTO 1510
1550 PRINT "                ";
1560 PRINT LN$(J)
1570 NEXT J:PRINT:GOTO 410
1580 FOR J=1 TO 40:PRINT:NEXT J:PRINT C7$:PRINT:PRINT
1590 PRINT                                              "
1600 PRINT " ";HI$;"            THE DUPECHECKER           "
1610 PRINT " ";HI$;"                                      "
1620 PRINT " ";HI$;"
1630 PRINT " ";HI$;"   (C) 1983 BY HAYDEN BOOK CO.,INC.   "
1640 PRINT " ";HI$;"
1650 PRINT LO$
1660 RETURN
1670 PRINT C7$:PRINT:PRINT
1680 PRINT HI$;"                                          "
1690 PRINT HI$;"              INSTRUCTIONS                "
1700 PRINT HI$;"                                          "
1710 PRINT HI$;"
1720 PRINT HI$;"  -ENTER CALL TO ADD IT TO DUPESHEET-     "
1730 PRINT HI$;"                                          "
1740 PRINT HI$;" 'N1NNN?' ENTER ANY CALL PLUS '?' FOR     "
1750 PRINT HI$;"        BAND-BY-BAND DUPECHECK            "
1760 PRINT HI$;"                                          "
1780 PRINT HI$;" 'X'    DELETES THE LAST CALL ENTERED     "
1790 PRINT HI$;" 'DEL'  DELETES ANY CALL                  "
1800 PRINT HI$;" 'QRT'  SAVES ALL CALLS AND EXITS         "
1810 PRINT HI$;" 'QSY'  CHANGES BANDS/DUPE SHEETS         "
1820 PRINT HI$;" 'SAVE' SAVES CALLS WITHOUT EXITING       "
1830 PRINT HI$;" '?'    DISPLAYS THIS MENU                "
1840 PRINT HI$;"                                          "
1850 PRINT LO$
1860 RETURN
1870 INPUT GS$:LL=LEN(GS$):IF LL=0 THEN 1930
1880 X1%=0:FOR J=1 TO LL:IF MID$(GS$,J,1)="-" THEN 510
1882 NEXT J
1890 LP=1
1900 IF LP>LL THEN 1930:REM 2ND ENTRY
1910 FOR LT=LP TO LL:IF MID$(GS$,LT,1)<>" " THEN 1940
1920 NEXT LT
1930 G$="":RETURN
1940 LP=LT
1950 FOR LT=LP TO LL:IF MID$(GS$,LT,1)=" " THEN 1980
1960 NEXT LT
1970 LT=LT+1
1980 G$=MID$(GS$,LP,LT-LP):LP=LT:RETURN
1990 REM PRINTER INIT, IF NEEDED
2000 RETURN
```

```
2010 PRINT
2020 PRINT HI$;"************************************"
2030 PRINT HI$;"** IT'S A DUPE  **  BOO  **  HISS **"
2040 PRINT HI$;"************************************"
2050 PRINT BL$:PRINT LO$:RETURN
2100 H%=E(4)+LG*26-26
2110 IF C1%(H%)=-32000 THEN RETURN
2120 IF C2%(H%)<>E2% THEN 2140
2130 IF C1%(H%)=E1% THEN GOSUB 2010:GOTO 1472
2140 IF P%(H%)=0 THEN RETURN
2150 H%=P%(H%):GOTO 2120
6000 INPUT#15,EA,EB$,EC,ED:IF EA<20 THEN RETURN:REM READ ERROR CHANNEL
6010 IF EA=50 THEN RETURN
6020 IF EA=63 THEN CLOSE4:OPEN4,8,4,"@0:DUPELOGA,S,W":RETURN
6030 PRINT EA;EB$,EC;ED:STOP:RETURN
7000 RH=INT(RN/256):RL=RN-256*RH:RETURN
7100 PRINT#15,"P"CHR$(3)CHR$(RL)CHR$(RH)CHR$(RP)
7110 GOTO 6000
```

A SAMPLE SESSION WITH THE DUPECHECKER

To illustrate how DUPECHECKER works, here's a step-by-step description of the way you might run a typical session with it, using an Apple II computer.

1. Before the contest begins, put a new initialized disk containing the DUPECHECKER and DUPEPRINT programs in the disk drive and run the DUPECHECKER. Ideally, the disk should be blank except for these programs to maximize the number of calls that can be stored.
2. The menu will come up on the screen. You'll be asked whether you're starting a new contest or resuming a previous operation. Type "N" for "new contest."
3. You will now be asked whether you want a hard-copy printout while entering calls. A hard copy is a good form of backup, just in case somebody should trip over the power cord. However, the hard copy isn't really needed for anything except backup. After the contest, The DUPEPRINT program will produce a nice alphabetized dupesheet, and you'll probably discard the on-line hard copy produced during the contest.
4. Next, enter the contest's name and the date, so the dupesheets will have the correct information.
5. You'll be asked how many bands you'll operate. Type "5" (you could specify any number from one band to ten, but let's make it five for now).
6. The DUPECHECKER will ask you to name each band. For this session, let's say the bands are "75," "40," "20," "15," and "10." The names can include letters as well as numbers, as in "10PH" (for phone) and

"10CW." This is needed for a contest like Field Day or the ARRL 10-meter contest, in which each station may be worked once on CW and again on phone. If this were a contest in which a station may be worked just once regardless of band, you'd specify one band, and perhaps just give it the name of the contest itself, as in "SS" for Sweepstakes.

7. You will now see a message telling you how many calls your computer's memory can accommodate. With an Apple II or 48K TRS-80, you can have well over 4,000 calls in memory at once.

8. At this point you'll be asked to specify a band. For instance, type "20."

9. After you specify a band, you're ready to begin entering calls. The main menu (shown in the accompanying photograph) will be displayed on the screen, along with a prompt ("DUPECHECKER READY FOR QSO #1 – 20").

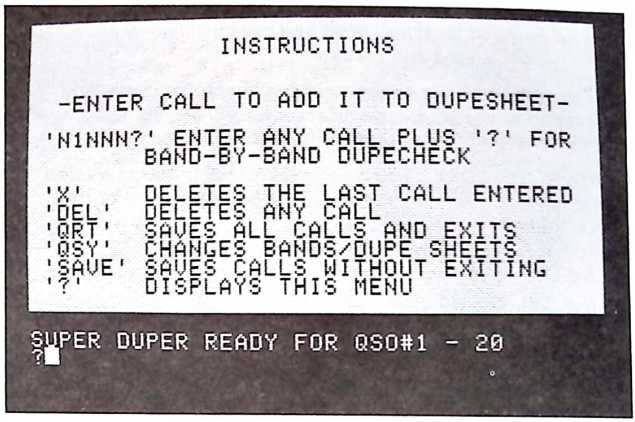
10. At this point, start typing calls. To enter a call, just type it with a carriage return. There is a delay of about ten seconds after you type the first call—but only after the initial call. From then on, the DUPECHECKER will accept and dupecheck calls very quickly. Even with 1,500 calls in memory, it typically takes less than two seconds to scan the list for dupes. If you enter a call that is a dupe, you *will* know it. The "IT'S-A-DUPE" message is designed to be conspicuous. Try re-entering a call that you've already entered once, just to see what happens.

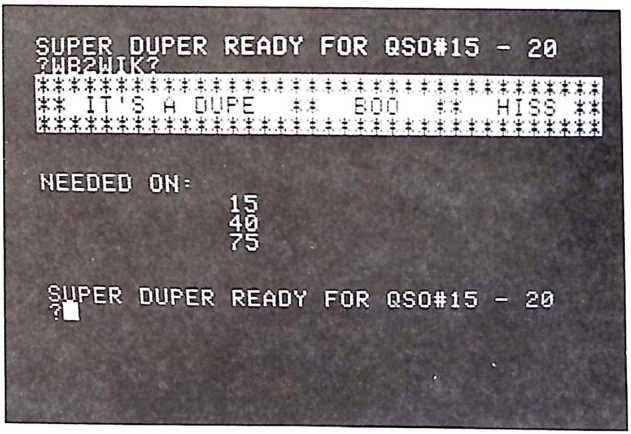
    The DUPECHECKER will accept any four- to six-character call sign, as long as the second or third digit (or both) is a number. The error-checking mechanism will tell you if a call is too short or too long, a feature that often catches typing errors.

    Since you're making only dupesheets and not logs at this point, you may omit portable designators if you wish. If you enter them, they will be disregarded for dupechecking purposes. And for the rare call that doesn't have four to six characters, simply add a character or two, perhaps an "X" and then a slash bar ("/") as a means of flagging the call as one that must be hand-corrected before the logs and dupesheets are submitted. For instance, if you're fortunate enough to work JY1 (King Hussein of Jordan, who exercised his monarchical prerogative and didn't assign himself a suffix), you might enter something like "JY1X/".

11. The main menu gives you a number of options besides just entering calls. For example, you can redisplay the menu itself by typing a question mark ("?") instead of a call sign.

12. To see what else besides rapid dupechecking this program can do to improve your contest score, try typing any call sign followed by a


INSTRUCTIONS
-ENTER CALL TO ADD IT TO DUPESHEET-
'N1NNN?' ENTER ANY CALL PLUS '?' FOR BAND-BY-BAND DUPECHECK
'X' DELETES THE LAST CALL ENTERED
'DEL' DELETES ANY CALL
'QRT' SAVES ALL CALLS AND EXITS
'QSY' CHANGES BANDS/DUPE SHEETS
'SAVE' SAVES CALLS WITHOUT EXITING
'?' DISPLAYS THIS MENU
SUPER DUPER READY FOR QSO#1 - 20
?



This is the DUPECHECKER Program. As the main menu explains (top photo), you simply type a call to enter each QSO. If it's a dupe, you'll be told (bottom photo). Or you can dupecheck a call without logging it as a completed QSO by typing the call followed by a question mark. Then you'll be told not only if it's a dupe but also if you still need the station on other bands (bottom photo).

question mark (e.g. "W3CCX?"). You will not only find out if the call is a dupe on the band you're on—you'll also see a list of other bands on which you still haven't worked that station! When you enter a call and a question mark, the call is only dupechecked, not entered into the log.

13. You also have other options. If you type "X" at the "DUPECHECKER READY" prompt, the call most recently entered will be deleted. That's

a command you'll get to know well if you're a typical contest operator. It's your bail-out function for times when you optimistically enter a call before the QSO is completed and confirmed.

14. To delete a call other than the one most recently entered, type "DEL". This is the function to use when you discover later that you had someone's call incorrect.

15. To change bands, type "QSY". You'll be asked to name the new band, and then the change is made immediately. Try entering some of the same calls you logged on the previous band. Or try entering a call you logged on another band, followed by a question mark. DUPECHECKER keeps track of these details nicely. To change bands even faster, you can type something like "QSY 15," placing the new band on the same line with the QSY command (with a space in between).

16. When the contest is over (or when you're taking a break), type "SAVE" or "QRT". Either of these functions saves all logged calls to disk. In order to assure the maximum possible execution speed, nothing goes to disk except when you type one of these two commands. In either case, the disk-access takes a couple of minutes. Plan accordingly.

17. When you're ready to print out your formal dupesheets after the contest, run the DUPEPRINT program, again using the disk that contains the data files created by the DUPECHECKER. Turn on the printer and tell DUPEPRINT whether you want an alphabetical sort (as you would in a DX contest) or a numerical sort by call areas (as you might in a domestic contest). Then the calls on each band will be sorted for printing. Be patient, though: sorting may take an hour or more if you have, say, 1,000 QSOs on one band. Just relax: go away and do something else, and check back periodically to see how things are going.

18. When it's all finished, you'll have a nicely organized set of dupesheets—and they'll be much easier to read and interpret than the old kind with hundreds (or thousands) of calls penciled in on a large and badly smudged sheet of paper.

THE DUPEPRINT PROGRAM

What it does DUPEPRINT reads the data files created by the DUPECHECKER or any of the contest logging programs and prepares well-organized dupesheets. The call signs may be sorted either alphabetically or by call area (with alphabetization within each call area).

Setup instructions As written, DUPEPRINT runs with standard 8½ × 11-inch paper (or 9½ × 11 fan-folded tractor-feed forms). Lines 70, 80 and 90 can be modified for different printing formats.

Another very worthwhile modification to consider is compiling DUPEPRINT with a BASIC compiler. With 1000 calls to sort, the sorting job alone can take an hour or more. If you have a compiler, such as the Hayden Applesoft Compiler for Apple computers or the Microsoft BASIC Compiler for CP/M systems, the sorting speed can be at least tripled. A much faster sorting algorithm could have been used, but we felt speed was less important in this particular program than doing the job without using additional RAM or disk space. Some computers will have little spare RAM or disk capacity by the time a major contest is logged on disk.

In the Apple version, the disk drive on which DUPEPRINT expects to find the data files can be changed from drive #1 to some other drive by modifying line 60. In the TRS-80 and CP/M version, the filenames in lines 1010 and 1070 could be changed to specify a disk drive other than the logged one, if desired. If you have modified the DUPECHECKER or LOGGER program to write the data files on a drive other than #1, you must modify this program similarly.

Operating notes

It doesn't take much to make DUPEPRINT run. Load and run it, making sure the data files are on the disk in the logged disk drive (the first drive unless the program is modified to specify some other drive).

The only thing you have to do to make the program operate is turn on the printer and tell the program whether to sort alphabetically or by call area. That's it.

DUPEPRINT will print a separate dupesheet for each band, except when the data files indicate otherwise. Thus, if the data files were generated by the SS LOGGER (or by either of the other loggers with only one band specified) you'll get one comprehensive dupesheet for the entire contest.

Modifications for CP/M

For CP/M systems running BASIC-80, modify the TRS-80 and CP/M listing as follows:

1. Delete the CLEAR statement in line 35;
2. Replace the clear-screen command in line 170 with the proper command for your system.

DUPEPRINT (TRS-80 and CP/M)

```
10 REM DUPE SHEET PRINTING PROGRAM (TRS-80)
20 REM BY KC6A AND N6NB
30 REM COPYRIGHT 1983 BY HAYDEN BOOK CO.,INC.
35 CLEAR 200
40 CLS:INPUT "WHAT IS YOUR CALL ";MY$
60 REM DATE FILES MUST BE ON LOGGED DRIVE (MOST SYSTEMS)
70 L1=66:REM L1 SETS # LINES/PAGE
80 L2=6: REM L2 SETS TOP AND BOTTOM MARGINS
90 L3=72:REM L3 SETS THE PAPER WIDTH IN PRINT POSITIONS
100 N1=0: REM N1 IS PAGE COUNTER
110 N2=L1+L2+1:REM N2 IS LINE COUNTER
120 N3=INT(L3/10):REM N3 IS COLUMNS/PAGE
130 DATA 1,2,3,4,5,6,36:REM SORT ORDER FOR ALPHABETIC SORTING
140 DATA 3,4,5,6,1,2,10:REM SORT ORDER FOR CALL AREA SORTING
150 PS=N3*(L1-2*L2-3-4):REM PAGE SIZE IN CALL SIGNS
160 DEF FNM(X)=X-L%*INT(X/L%)
161 DEF FNX(X)=FNM(X+FP-1)
170 CLS:REM CLEAR SCREEN
180 PRINT:PRINT
```

```
190 PRINT TAB(18)"DUPE SHEET PRINTING PROGRAM"
200 PRINT:PRINT
210 GOSUB 1000
220 PRINT
230 INPUT " SHOULD CALLS BE LISTED ALPHABETICALLY OR BY CALL AREA (A/C)";A$
240 PRINT
250 IF A$="C" THEN GOSUB 1100 ELSE IF A$<>"A" THEN 220
260 GOSUB 1100
270 INPUT "POSITION PAPER AT TOP OF A NEW PAGE IN PRINTER. THEN <CR>";G$
280 DIM QC(ST),X1%(100),X2%(100)
290 AV=ASC("0"):AS=26:IF A$="A" THEN AV=ASC("A"):AS=0
300 FOR BD=1 TO NL
310 PRINT:PRINT "PROCESSING DUPESHEET ";LN$(BD):PRINT
320 FOR KL=1 TO ST
330 IF KL=1 THEN FOR J=1 TO ST:QC(J)=0:NEXT J
340 MP=0:LP=0:KT=KL+AS:FP=1:LR=0
350 W$=CHR$(KL+AV-1):IF KL>26 THEN W$=CHR$(KL-27+ASC("0"))
360 PRINT:PRINT "   SORTING BY... "W$
370 IF KL>1 THEN IF QC(KL)<1 THEN 900
380 FOR K=1 TO 26
390 H%=BD*26-26+K
400 GET 2,H%
410 E1%=CVI(C1$):E2%=CVI(C2$):PZ%=CVI(PZ$)
420 LR=LR+1:IF LR=10 THEN LR=0:PRINT "+";
430 IF E1%=-32000 THEN 650:REM NO ENTRY
440 IF E1%<-32000 THEN 640:REM BLANK ENTRY
450 GOSUB 1160
460 IF A$<>"A" THEN 490
470 IF E(1)<>0 THEN 490
480 E(1)=E(2):E(2)=0
490 QC(E(S(1))-AS)=1:IF E(S(1))<>KT THEN GOTO 640:REM SKIP
500 E1%=-32000+E(S(1))*1369+E(S(2))*37+E(S(3))
510 E2%=-32000+E(S(4))*1369+E(S(5))*37+E(S(6))
520 REM SORT INTO ARRAY LIST
530 SI=0
540 IF SI>=MP THEN 620
550   SI=SI+1
560   IF E1%<X1%(SI) THEN 590
570   IF E1%>X1%(SI) THEN 610
580   IF E2%>X2%(SI) THEN 610
590   DX%=X1%(SI):X1%(SI)=E1%:E1%=DX%
600   DY%=X2%(SI):X2%(SI)=E2%:E2%=DY%
610 GOTO 540
620 MP=MP+1:X1%(MP)=E1%:X2%(MP)=E2%
630 IF MP=100 THEN GOSUB 1440:REM MERGE
640 IF PZ%<>0 THEN H%=PZ%:GOTO 400
650 NEXT K
660 IF MP>0 THEN GOSUB 1440
670 REM RECONSTRUCT CALL & PRINT IT
680 IF QC(KL)=0 THEN 900:REM LIST EMPTY
690 SE=LP-FP+1:IF SE<0 THEN SE=SE+L%
700 FOR JJ=SE+1 TO N3*INT((SE+N3-1)/N3)
710 C1%(FNX(JJ))=-32000:C2%(FNX(JJ))=-32000:SE=SE+1
720 NEXT JJ
730 IF N2+3+SE/N3>L1-L2 THEN GOSUB 1330
740 PL=1
750 PU=SE:IF SE>PS+PL-1 THEN PU=PS+PL-1
760 LPRINT:LPRINT TAB(L3/2-1)"-";W$;"-":LPRINT:N2=N2+3
770 FOR JJ=PL TO PL-1+(PU-PL+1)/N3
780 FOR KK=0 TO N3-1
790 SS=FNX(INT(JJ+KK*(PU-PL+1)/N3))
800 EX=C1%(SS)+32000:EY=C2%(SS)+32000:GOSUB 1180
810 GOSUB 1230
820 LPRINT "  ";P$;"  ";
830 NEXT KK
840 LPRINT:N2=N2+1
850 NEXT JJ
860 IF PU>=SE THEN 900
870 PL=PU+1
880 GOSUB 1330
```

```
890 GOTO 750
900 NEXT KL
910 REM SPACE TO END OF PAGE
920 FOR Z=N2+1 TO L2+L1
930 LPRINT
940 NEXT Z
950 N2=L1+L2
960 NEXT BD
970 PRINT:PRINT TAB(5)"-DONE"
980 CLOSE
990 END
1000 REM LOAD DUPE FILE
1010 OPEN "I",1,"DUPELOGA"
1020 INPUT#1,HC$:INPUT#1,T:INPUT#1,Y$:INPUT#1,Z$:INPUT#1,NL:INPUT#1,L%:INPUT#1,M
     1
1030 L%=L%+10
1040 DIM LN$(NL),C1%(L%),C2%(L%)
1050 FOR J=1 TO NL:INPUT#1,LN$(J):NEXT J
1060 CLOSE
1070 OPEN "R",2,"DUPELOGB",6
1080 FIELD 2,2 AS C1$,2 AS C2$,2 AS PZ$
1090 RETURN
1100 FOR J=1 TO 6
1110 READ S(J)
1120 T(S(J))=J
1130 NEXT J
1140 READ ST
1150 RETURN
1160 REM DECODE CALL INTO 6 DIGITS
1170 EX=E1%+32000:EY=E2%+32000
1180 E(1)=INT(EX/1369):EX=EX-E(1)*1369
1190 E(2)=INT(EX/37):E(3)=EX-E(2)*37
1200 E(4)=INT(EY/1369):EY=EY-E(4)*1369
1210 E(5)=INT(EY/37):E(6)=EY-E(5)*37
1220 RETURN
1230 REM CHANGE CALLS TO ALPHA
1240 IF A$="A" THEN IF E(2)=0 THEN E(2)=E(1):E(1)=0
1250 P$=" "
1260 FOR L=1 TO 6
1270 LL=E(T(L))
1280 IF LL=0 THEN P$=P$+" ":GOTO 1310
1290 IF LL<27 THEN P$=P$+CHR$(LL+64):GOTO 1310:REM LETTER
1300 P$=P$+CHR$(LL+21):REM NUMBER
1310 NEXT L
1320 RETURN
1330 REM GO TO NEW PAGE
1340 FOR Z=N2+1 TO L2+L1
1350 LPRINT
1360 NEXT Z
1370 N1=N1+1
1380 LPRINT TAB(5)Z$;
1390 LPRINT TAB((L3-LEN(MY$))/2)MY$;
1400 LPRINT TAB(L3-8)"PAGE ";N1
1410 LPRINT TAB((L3-LEN(Y$))/2)Y$
1420 LPRINT TAB((L3-10-LEN(LN$(BD)))/2)"DUPESHEET ";LN$(BD)
1430 LPRINT:N2=L2+4:RETURN
1440 SI=1:CP=FP:NP=LP:REM MERGE
1450 IF CP=FNM(LP+1) THEN IF SI>MP THEN 1550
1460   NP=FNM(NP+1)
1470   IF SI>MP THEN 1530
1480   IF CP=FNM(LP+1) THEN 1520
1490   IF X1%(SI)<C1%(CP)THEN 1520
1500   IF X1%(SI)>C1%(CP) THEN 1530
1510   IF X2%(SI)>C2%(CP) THEN 1530
1520   C1%(NP)=X1%(SI):C2%(NP)=X2%(SI):SI=SI+1:GOTO 1540
1530   C1%(NP)=C1%(CP):C2%(NP)=C2%(CP):CP=FNM(CP+1)
1540 GOTO 1450
1550 FP=FNM(LP+1)
1560 LP=NP:MP=0:RETURN
```

DUPEPRINT (Apple)

```
10  REM  DUPE SHEET PRINT PROGRAM
20  REM  BY KC6A AND N6NB
40  HOME : INPUT "WHAT IS YOUR CALL?";MY$
50 D$ =  CHR$ (4):LP$ = D$ + "PR#1":SCN$ = D$ + "PR#0"
60 DV$ = ",D1": REM  DRIVE FOR DATA FILES
62 RD$ = D$ + "READ "
63 CL$ = D$ + "CLOSE ":OP$ = D$ + "OPEN "
64 R2$ = ",L119":F2$ = "DUPELOGB"
66 F1$ = "DUPELOGA"
70 L1 = 66: REM  L1 SETS # LINES/PAGE
80 L2 = 6
90 L3 = 72: REM  L3 SETS PAPER WIDTH IN PRINT POSITIONS
100 N1 = 0: REM  N1 IS PAGE COUNTER
110 N2 = L1 + L2 + 1: REM  N2 IS LINE COUNTER
120 N3 =  INT (L3 / 10): REM  N3 IS COLUMNS/PAGE
130  DATA  1,2,3,4,5,6,36: REM  SORT ORDER FOR ALPHABETIC SORTING
140  DATA  3,4,5,6,1,2,10: REM  SORT ORDER FOR CALL AREA SORTING
150 PS = N3 * (L1 - 2 * L2 - 3 - 4): REM  PAGE SIZE IN CALL SIGNS
160  GOSUB 1090
161  DEF  FN M(X) = X - L% *  INT (X / L%)
162  DEF  FN X(X) =  FN M(X + FP - 1)
170  HOME
180  PRINT : INVERSE
188  PRINT "                                        "
190  PRINT "                                        "
192  PRINT "    DUPE SHEET PRINTING PROGRAM         "
194  PRINT "  (C) 1983 BY HAYDEN BOOK CO.,INC.      "
196  PRINT "                                        "
200  NORMAL
210  PRINT : PRINT
220  GOSUB 990
230  PRINT
232  PRINT "SHOULD CALLS BE LISTED ALPHABETICALLY"
240  INPUT "  OR BY CALL AREA (A/C)? ";A$
250  PRINT
255  IF A$ <  > "C" THEN  IF A$ <  > "A" THEN 220
270  IF A$ = "C" THEN  GOSUB 1090
272  PRINT "POSITION PAPER AT TOP OF A NEW"
280  INPUT "PAGE IN PRINTER. THEN <CR>";G$
290  DIM QC(ST),X1%(100),X2%(100)
300 AV =  ASC ("0"):AS = 26: IF A$ = "A" THEN AV =  ASC ("A"):AS = 0
310  FOR BD = 1 TO NL
320  VTAB (15): PRINT "     PROCESSING DUPESHEET ";LN$(BD)
330  FOR KL = 1 TO ST
340  IF KL = 1 THEN  FOR J = 1 TO ST:QC(J) = 0: NEXT J
345 MP = 0:LP = 0:KT = KL + AS:FP = 1
350 W$ =  CHR$ (KL + AV - 1): IF KL > 26 THEN W$ =  CHR$ (KL - 27 +  ASC ("0"))
360  VTAB (17): PRINT "     ..SORTING BY... "W$
370  IF KL > 1 THEN  IF QC(KL) < 1 THEN 890
380  FOR K = 1 TO 26
390 H% = BD * 26 - 26 + K
392  PRINT OP$;F2$;R2$;DV$
400  PRINT RD$;F2$;",R";H%
410  INPUT E1%: INPUT E2%: INPUT PZ%
420  PRINT CL$: VTAB (19): PRINT "     READING REC# ";H%;"      "
430  IF E1% =  - 32000 THEN 640: REM  NO ENTRY
440  IF E1% <  - 32000 THEN 630: REM  BLANK ENTRY
450  GOSUB 1150
460  IF A$ <  > "A" THEN 480
470  IF E(1) <  > 0 THEN 480
480 E(1) = E(2):E(2) = 0
490 QC(E(S(1)) - AS) = 1: IF E(S(1)) <  > KT THEN  GOTO 630: REM  SKIP
500 E1% =  - 32000 + E(S(1)) * 1369 + E(S(2)) * 37 + E(S(3))
510 E2% =  - 32000 + E(S(4)) * 1369 + E(S(5)) * 37 + E(S(6))
520  REM  SORT INTO ARRAY LIST
530 SI = 0
540  IF SI >  = MP THEN 610
550 SI = SI + 1
```

```
550  IF E1% < X1%(SI) THEN 580
560  IF E1% > X1%(SI) THEN 600
570  IF E2% > X2%(SI) THEN 600
580 DX% = X1%(SI):X1%(SI) = E1%:E1% = DX%
590 DY% = X2%(SI):X2%(SI) = E2%:E2% = DY%
600  GOTO 530
610 MP = MP + 1:X1%(MP) = E1%:X2%(MP) = E2%
620  IF MP = 100 THEN  GOSUB 1430: REM  MERGE
630  IF PZ% <  > 0 THEN H% = PZ%: GOTO 390
640  NEXT K
642  PRINT CL$
650  IF MP > 0 THEN  GOSUB 1430
660  REM  RECONSTRUCT CALL & PRINT IT
661  VTAB (19): PRINT "          PRINTING          "
662  PRINT LP$
670  IF QC(KL) = 0 THEN 890: REM  LIST EMPTY
680 SE = LP - FP + 1: IF SE < 0 THEN SE = SE + L%
685  IF SE + 1 > N3 *  INT ((SE + N3 - 1) / N3) THEN 720
690  FOR JJ = SE + 1 TO N3 *  INT ((SE + N3 - 1) / N3)
700 C1%( FN X(JJ)) =  - 32000:C2%( FN X(JJ)) =  - 32000:SE = SE + 1
710  NEXT JJ
720  IF N2 + 3 + SE / N3 > L1 - L2 THEN  GOSUB 1320
730 PL = 1
740 PU = SE: IF SE > PS + PL - 1 THEN PU = PS + PL - 1
750  PRINT : PRINT  SPC( L3 / 2 - 2)"-";W$;"-": PRINT :N2 = N2 + 3
760  FOR JJ = PL TO PL - 1 + (PU - PL + 1) / N3
770  FOR KK = 0 TO N3 - 1
780 SS =  FN X( INT (JJ + KK * (PU - PL + 1) / N3))
790 EX = C1%(SS) + 32000:EY = C2%(SS) + 32000: GOSUB 1170
800  GOSUB 1220
810  PRINT "  ";P$;"  ";
820  NEXT KK
830  PRINT :N2 = N2 + 1
840  NEXT JJ
850  IF PU >  = SE THEN 890
860 PL = PU + 1
870  GOSUB 1320
880  GOTO 740
890  PRINT SCN$: NEXT KL
900  REM  SPACE TO END OF PAGE
905  PRINT LP$
910  FOR Z = N2 + 1 TO L2 + L1
920  PRINT
930  NEXT Z: PRINT SCN$
940 N2 = L1 + L2
950  NEXT BD
960  PRINT SCN$: VTAB (21): PRINT  TAB( 10)"-DONE-"
970  PRINT CL$
980  END
990  REM  LOAD DUPE FILE
1000  PRINT OP$;F1$;DV$
1005  PRINT RD$;F1$
1010  INPUT HC$: INPUT T: INPUT Y$: INPUT Z$: INPUT NL: INPUT L%: INPUT M1
1020 L% = L% + 10
1030  DIM LN$(NL),C1%(L%),C2%(L%)
1040  FOR J = 1 TO NL: INPUT LN$(J): NEXT J
1050  PRINT CL$
1080  RETURN
1090  FOR J = 1 TO 6
1100  READ S(J)
1110 T(S(J)) = J
1120  NEXT J
1130  READ ST
1140  RETURN
1150  REM  DECODE CALL INTO 6 DIGITS
1160 EX = E1% + 32000:EY = E2% + 32000
1170 E(1) =  INT (EX / 1369):EX = EX - E(1) * 1369
1180 E(2) =  INT (EX / 37):E(3) = EX - E(2) * 37
1190 E(4) =  INT (EY / 1369):EY = EY - E(4) * 1369
```

```
1200 E(5) =  INT (EY / 37):E(6) = EY - E(5) * 37
1210  RETURN
1220  REM  CHANGE CALLS TO ALPHA
1230  IF A$ = "A" THEN  IF E(2) = 0 THEN E(2) = E(1):E(1) = 0
1240 P$ = " "
1250  FOR L = 1 TO 6
1260 LL = E(T(L))
1270  IF LL = 0 THEN P$ = P$ + " ": GOTO 1300: REM  SPACE
1280  IF LL < 27 THEN P$ = P$ +  CHR$ (LL + 64): GOTO 1300: REM  LETTER
1290 P$ = P$ +  CHR$ (LL + 21): REM  NUMBER
1300  NEXT L
1310  RETURN
1320  REM  GO TO NEW PAGE
1330  IF N2 + 1 > L2 + L1 THEN 1360
1335  FOR Z = N2 + 1 TO L2 + L1
1340  PRINT
1350  NEXT Z
1360 N1 = N1 + 1
1370  PRINT  SPC( 4)Z$;
1380  PRINT  SPC( (L3 -  LEN (MY$)) / 2 - 5 -  LEN (Z$))MY$;
1390  PRINT  SPC( (L3 -  LEN (MY$)) / 2 - 12)"PAGE";N1
1400  PRINT  SPC( (L3 -  LEN (Y$)) / 2)Y$
1410  PRINT  SPC( (L3 - 10 -  LEN (LN$(BD))) / 2)"DUPESHEET ";LN$(BD)
1420  PRINT :N2 = L2 + 4: RETURN
1430  REM  MERGE
1440 SI = 1:CP = FP:NP = LP
1450  IF CP =  FN M(LP + 1) THEN  IF SI > MP THEN 1550
1460 NP =  FN M(NP + 1)
1470  IF SI > MP THEN 1530
1480  IF CP =  FN M(LP + 1) THEN 1520
1490  IF X1%(SI) < C1%(CP) THEN 1520
1500  IF X1%(SI) > C1%(CP) THEN 1530
1510  IF X2%(SI) > C2%(CP) THEN 1530
1520 C1%(NP) = X1%(SI):C2%(NP) = X2%(SI):SI = SI + 1: GOTO 1540
1530 C1%(NP) = C1%(CP):C2%(NP) = C2%(CP):CP =  FN M(CP + 1)
1540  GOTO 1450
1550 FP =  FN M(LP + 1)
1560 LP = NP:MP = 0: RETURN
```

DUPEPRINT (Commodore 64)

```
 5 REM
 10 REM DUPE SHEET PRINTING PROGRAM (C64 VERSION)
 20 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
 30 OPEN 130,2,2,CHR$(136):REM RS232 PRINTER CHANNEL
 33 OPEN15,8,15
 50 C7$=CHR$(147):REM PUT YOUR CLEAR SCREEN COMMAND HERE
 70 L1=66:REM L1 SETS # LINES/PAGE
 80 L2=6: REM L2 SETS TOP AND BOTTOM MARGINS
 90 L3=72:REM L3 SETS THE PAPER WIDTH IN PRINT POSITIONS
 100 N1=0: REM N1 IS PAGE COUNTER
 110 N2=L1+L2+1:REM N2 IS LINE COUNTER
 120 N3=INT(L3/10):REM N3 IS COLUMNS/PAGE
 130 DATA 1,2,3,4,5,6,36:REM SORT ORDER FOR ALPHABETIC SORTING
 140 DATA 3,4,5,6,1,2,10:REM SORT ORDER FOR CALL AREA SORTING
 150 PS=N3*(L1-2*L2-3-4):REM PAGE SIZE IN CALL SIGNS
 161 DEF FNM(X)=X-L%*INT(X/L%)
 162 DEF FNX(X)=FNM(X+FP-1)
 170 PRINT C7$:REM CLEAR SCREEN
 180 PRINT:PRINT
 190 PRINT "       DUPE SHEET PRINTING PROGRAM"
 200 PRINT:PRINT
 204 INPUT "WHAT IS YOUR CALL";MY$
 210 GOSUB 1000
 220 PRINT
```

```
230 PRINT "SHOULD CALLS BE LISTED ALPHABETICALLY"
232 INPUT "OR BY CALL AREA (A/C)";A$
240 PRINT
250 IF A$="C" THEN GOSUB 1100 ELSE IF A$<>"A" THEN 220
260 GOSUB 1100
270 PRINT "POSITION PAPER AT TOP OF A NEW PAGE"
272 INPUT "IN PRINTER. THEN <CR>";G$
280 DIM QC(SZ), X1%(100), X2%(100)
290 AV=ASC("0"):AS=26:IF A$="A" THEN AV=ASC("A"):AS=0
300 FOR BD=1 TO NL
310 PRINT:PRINT "PROCESSING DUPESHEET ";LN$(BD):PRINT
320 FOR KL=1 TO SZ
330 IF KL=1 THEN FOR J=1 TO SZ:QC(J)=0:NEXT J
340 MP=0:LP=0:KT=KL+AS:FP=1:LR=0
350 W$=CHR$(KL+AV-1):IF KL>26 THEN W$=CHR$(KL-27+ASC("0"))
360 PRINT:PRINT "   SORTING BY... "W$
370 IF KL>1 THEN IF QC(KL)<1 THEN 900
380 FOR K=1 TO 26
390 H%=BD*26-26+K
400 RN=H%:GOSUB7000
410 RP=1:GOSUB7100:INPUT#3,E1%
412 RP=9:GOSUB7100:INPUT#3,E2%
414 RP=17:GOSUB7100:INPUT#3,PZ%
420 LR=LR+1:IF LR=10 THEN LR=0:PRINT "+";
430 IF E1%=-32000 THEN 650:REM NO ENTRY
440 IF E1%<-32000 THEN 640:REM BLANK ENTRY
450 GOSUB 1160
460 IF A$<>"A" THEN 490
470 IF E(1)<>0 THEN 490
480 E(1)=E(2):E(2)=0
490 QC(E(S(1))-AS)=1:IF E(S(1))<>KT THEN GOTO 640:REM SKIP
500 E1%=-32000+E(S(1))*1369+E(S(2))*37+E(S(3))
510 E2%=-32000+E(S(4))*1369+E(S(5))*37+E(S(6))
520 REM SORT INTO ARRAY LIST
530 SI=0
540 IF SI>=MP THEN 620
550 SI=SI+1
560 IF E1%< X1%(SI) THEN 590
570 IF E1%> X1%(SI) THEN 610
580 IF E2%> X2%(SI) THEN 610
590 DX%= X1%(SI): X1%(SI)=E1%:E1%=DX%
600 DY%= X2%(SI): X2%(SI)=E2%:E2%=DY%
610 GOTO 540
620 MP=MP+1: X1%(MP)=E1%: X2%(MP)=E2%
630 IF MP=100 THEN GOSUB 1440:REM MERGE
640 IF PZ%<>0 THEN H%=PZ%:GOTO 400
650 NEXT K
660 IF MP>0 THEN GOSUB 1440
670 REM RECONSTRUCT CALL & PRINT IT
680 IF QC(KL)=0 THEN 900:REM LIST EMPTY
690 SE=LP-FP+1:IF SE<0 THEN SE=SE+L%
700 FOR JJ=SE+1 TO N3*INT((SE+N3-1)/N3)
710 C1%(FNX(JJ))=-32000:C2%(FNX(JJ))=-32000:SE=SE+1
720 NEXT JJ
730 IF N2+3+SE/N3>L1-L2 THEN GOSUB 1330
740 PL=1
750 PU=SE:IF SE>PS+PL-1 THEN PU=PS+PL-1
760 PRINT#130,"":PRINT#130,"";SPC(L3/2-1)"-";W$;"-":PRINT#130,"":N2=N2+3
770 FOR JJ=PL TO PL-1+(PU-PL+1)/N3
780 FOR KK=0 TO N3-1
790 SS=FNX(INT(JJ+KK*(PU-PL+1)/N3))
800 EX=C1%(SS)+32000:EY=C2%(SS)+32000:GOSUB 1180
810 GOSUB 1230
820 PRINT#130,"  ";P$;"  ";
830 NEXT KK
840 PRINT#130,"":N2=N2+1
850 NEXT JJ
860 IF PU>=SE THEN 900
870 PL=PU+1
```

```
880 GOSUB 1330
890 GOTO 750
900 NEXT KL
910 REM SPACE TO END OF PAGE
920 FOR Z=N2+1 TO L2+L1
930 PRINT#130,""
940 NEXT Z
950 N2=L1+L2
960 NEXT BD
970 PRINT:PRINT TAB(5)"-DONE"
980 CLOSE3:CLOSE15
990 FOR J=1 TO 1500:PRINT;:NEXT J:CLOSE130:END
1000 REM LOAD DUPE FILE
1010 OPEN4,8,4,"0:DUPELOGA,S,R":GOSUB6000
1020 INPUT#4,HC$:INPUT#4,T:INPUT#4,Y$:INPUT#4,Z$:INPUT#4,NL
1021 INPUT#4,L%:INPUT#4,M1
1030 L%=L%+10
1040 DIM LN$(NL),C1%(L%),C2%(L%)
1050 FOR J=1 TO NL:INPUT#4,LN$(J):NEXT J
1060 CLOSE4
1070 OPEN3,8,3,"0:DUPELOGB,L,"+CHR$(23):GOSUB6000
1090 RETURN
1100 FOR J=1 TO 6
1110 READ S(J)
1120 T(S(J))=J
1130 NEXT J
1140 READ SZ
1150 RETURN
1160 REM DECODE CALL INTO 6 DIGITS
1170 EX=E1%+32000:EY=E2%+32000
1180 E(1)=INT(EX/1369):EX=EX-E(1)*1369
1190 E(2)=INT(EX/37):E(3)=EX-E(2)*37
1200 E(4)=INT(EY/1369):EY=EY-E(4)*1369
1210 E(5)=INT(EY/37):E(6)=EY-E(5)*37
1220 RETURN
1230 REM CHANGE CALLS TO ALPHA
1240 IF A$="A" THEN IF E(2)=0 THEN E(2)=E(1):E(1)=0
1250 P$=" "
1260 FOR L=1 TO 6
1270 LL=E(T(L))
1280 IF LL=0 THEN P$=P$+" ":GOTO 1310
1290 IF LL<27 THEN P$=P$+CHR$(LL+64):GOTO 1310:REM LETTER
1300 P$=P$+CHR$(LL+21):REM NUMBER
1310 NEXT L
1320 RETURN
1330 REM GO TO NEW PAGE
1340 FOR Z=N2+1 TO L2+L1
1350 PRINT#130,""
1360 NEXT Z
1370 N1=N1+1
1380 PRINT#130,"     ";Z$;
1390 PRINT#130,"";SPC((L3-LEN(MY$))/2-5-LEN(Z$))MY$;
1400 PRINT#130,"";SPC((L3-LEN(MY$))/2-12)"PAGE";N1
1410 PRINT#130,"";SPC((L3-LEN(Y$))/2)Y$
1420 PRINT#130,"";SPC((L3-9-LEN(LN$(BD)))/2)"DUPESHEET ";LN$(BD)
1430 PRINT#130,"":N2=L2+4:RETURN
1440 REM MERGE
1450 SI=1:CP=FP:NP=LP
1460 IF CP=FNM(LP+1) THEN IF SI>MP THEN 1560
1470 NP=FNM(NP+1)
1480 IF SI>MP THEN 1540
1490 IF CP=FNM(LP+1) THEN 1530
1500 IF  X1%(SI)<C1%(CP)THEN 1530
1510 IF  X1%(SI)>C1%(CP) THEN 1540
1520 IF  X2%(SI)>C2%(CP) THEN 1540
1530 C1%(NP)=X1%(SI):C2%(NP)=X2%(SI):SI=SI+1:GOTO 1550
1540 C1%(NP)=C1%(CP):C2%(NP)=C2%(CP):CP=FNM(CP+1)
1550 GOTO 1460
1560 FP=FNM(LP+1)
1570 LP=NP:MP=0:RETURN
```

```
6000 INPUT#15,EA,EB$,EC,ED:IF EA<20 THEN RETURN:REM READ ERROR CHANNEL
6030 PRINT EA;EB$,EC;ED:STOP:RETURN
7000 RH=INT(RN/256):RL=RN-256*RH:RETURN
7100 PRINT#15,"P"CHR$(3)CHR$(RL)CHR$(RH)CHR$(RP)
7110 GOTO 6000
```

THE CONTEST LOGGING PROGRAMS

What they do

Each of the three programs in this package is a complete, full-featured contest logger. Each includes all of the dupechecking features of the DUPECHECKER plus logging, scoring, and editing capabilities.

Each logging program will give you an instant display of your running total score (and a list of the multipliers you've worked on each band) at any time during or after the contest. You can do a band-by-band dupecheck on any call sign, quickly spotting any bands on which you still need that station.

One of the best features of the contest loggers is their editing capability. If you discover you've made a mistake in logging any previous contact, it's easy to fix. You can quickly revise any logged QSO and, yes, the logger recomputes your score and multiplier tally after each edit. If you discover your one and only contact in a rare multiplier was really someplace else, the logger will deduct that multiplier from your total. And, of course, if the editing increases your score or multiplier, the logger programs take care of that change, too.

When it's all over, the LOGPRINT program goes to work, producing a neat, well-organized set of contest logs for you. The DUPEPRINT program will prepare your dupesheets, arranging all calls as you specify: alphabetically or by call areas.

The GENERAL PURPOSE CONTEST LOGGER works (and keeps correct scores) for any contest in which the scoring is on a QSO-points-times-multipliers basis, with all or part of the exchange constituting the multiplier. You can choose whether to count band-multipliers or just count each multiplier once in the whole contest. Likewise, you can set it up to do the dupechecking on a band-by-band basis (for contests in which you can work each station once per band) or on an overall basis (for contests in which you can only work a given station once, regardless of band).

The GENERAL CONTEST LOGGER can be used in a wide variety of contests, ranging from VHF-UHF contests and QSO parties to the ARRL DX contest. It is suited for many of the specialty contests as well. The contest logging package also includes customized programs for two very popular contests: ARRL Sweepstakes and Field Day.

The FIELD DAY LOGGER is tailored to the unique scoring and operating procedures of Field Day, which is surely the most popular ham radio operating activity in North America. The Field Day scoring system is unusual in that the multipliers are not based on geography. The FD LOGGER understands the special rules of Field Day, and it offers the same dupechecking, editing, and quick-summary provisions of the GENERAL CONTEST LOGGER, but with custom features for Field Day.

Field Day is more than a contest, of course. It's also a group outing, and sometimes a weekend-long party. But for those who want to keep track of the operating aspect of Field Day, this logger is ideal. Because almost every radio club has at least one member who owns a computer on which the FD LOGGER will run, this program offers an ideal opportunity to combine hamming with personal computing.

Sweepstakes is another contest with specialized rules. It has its own exchange, and a rather long one at that. It also has a fixed multiplier structure: no matter how many bands you operate, you can only contact a fixed number of multipliers—and you can only count them once regardless of band. The SS LOGGER is customized for these unique features: it even has something of a "clean sweep" checker that displays all possible multipliers and flags the ones you have. Every year, a number of the top operators work a "clean sweep," picking up every possible multiplier.

Setup instructions

The loggers all have one hardware-dependent special feature: a provision for a real-time clock.

If you have a real-time clock in your computer, these loggers will take advantage of that feature to do automatic time-keeping. Until you've run a contest with automatic time-keeping, you may not realize just what a great convenience it can be.

The Apple version of all three logger programs is written for the Mountain Computers CPS Multifunction Card, which includes serial and parallel interfaces as well as a dependable clock, in slot #1. If you have a different clock, you may need to change the time-input routine at line 1162 in the GENERAL LOGGER (line 1182 in both the SS LOGGER and the FD LOGGER).

With this clock card installed, an Apple computer will do accurate and automatic time-keeping for you. Of course, if you're retyping a set of handwritten logs after a contest, you wouldn't want real time entered as you type the information for each contact. At the very beginning of a session with any of the loggers, you can tell the program not to use the real-time clock routine, instead asking you the time for each QSO. (You can also hit a carriage return to skip the time if you don't want it listed for every QSO.)

If you do not have a real-time clock, you need not modify the programs. Simply tell the logger not to use the real-time clock each time you start up, and there will be no problems.

The TRS-80 and CP/M version of the program is written so that it will input the time from the built-in clock in a TRS-80 Model III. That clock, which takes its timing from the system clock, leaves something to be desired in accuracy, since it stops momentarily during disk access. It loses 5–6 minutes during a typical day's logging. Thus, if you use it, we suggest you set the time just before the contest, and reset it at each break. "Rubber clocking" is a serious problem in contesting, and you wouldn't want your computer doing it.

As with the Apple, if you're using a different clock, you may have to write a different clock routine. Like the Apple version, the TRS-80 and CP/M version has this routine at line 1162 in the GENERAL LOGGER and 1182 in both the SS LOGGER and FD LOGGER.

The Commodore 64 has what the manufacturer calls a "jiffy clock," which also takes its time from the system clock. Like the TRS-80 clock, it stops during some I/O access. The Commodore version of the GENERAL PURPOSE LOGGER uses this clock for logging. But because this clock is not accurate in the long run, the listing includes a provision to reset the clock at regular intervals. As you use the program, you will be automatically asked to enter the correct time every 50 QSOs. This is far more convenient than having to enter the time for every QSO, and it still provides sufficient accuracy for most contests.

TRS-80 Model III users could add a similar feature to the logging programs. A routine to do this on a Model III, written for the GENERAL LOGGER, is shown here:

```
455 IF T/50=INT(T/50) THEN GOSUB 4210
```

A similar subroutine for the SS LOGGER or FD LOGGER would begin at line 515 instead of 455. Nothing else would need to be changed.

Although this subroutine will work only with the TRS-80, something similar could be written for any computer whose real-time clock can be accessed from BASIC. Chapter 5 explains how to modify this syntax for an IBM PC.

If you're using one of the computers for which these programs were written, no other hardware-dependent modifications should be needed, except possibly to route the data files to something other than the primary (or logged) disk drive. Line 58 in all three Apple loggers sets the variable DV$ to ",D1". If you want your data files on drive 2, change it to ",D2". In the TRS-80 and CP/M version, a drive designation could be added to the data filenames, which appear in line 70 of each logger listing.

If you have more than one disk drive, this is an important modification. The data file called "CONTSTLG" will be by far the largest, and it should be placed on a disk all by itself if you have a second disk drive.

Commodore 64 users will note that only the GENERAL CONTEST LOGGER for the Commodore is listed in this chapter. Space limitations precluded reproducing all three loggers in three different versions. However, if you compare the Commodore GENERAL LOGGER with the TRS-80 version you can see the syntax changes that are required. The other loggers can be modified for the Commodore by making equivalent syntax changes to the TRS-80 listing.

Operating notes

The operating procedures for the loggers are discussed elsewhere in this chapter. However, a few things should be pointed out here.

Like the DUPECHECKER, the various loggers store call signs in RAM for rapid dupechecking. But they also write log information on disk at regular intervals. This is necessary to keep the RAM free for the call signs. Thus, the log size is limited by disk storage capacity. On an Apple II with a single disk drive, the capacity is about 1200 QSOs. An Apple with two drives—and with the primary contest log written to the second drive—can accommodate about 1800 contacts. On systems with greater disk storage capacity, far more contacts can be logged, of course.

It takes an Apple II about five seconds to write a QSO on disk, which means you can log up to about 12 QSOs a minute to disk if necessary. Since even the best contest operators rarely make more than five QSOs per minute, the delay here should not be a problem. Some CP/M-based systems write data to disk much faster than this. Also, the Commodore, TRS-80, and CP/M versions of the loggers write QSO data to disk only when the random access buffer is full (usually each five or six contacts), not after every contact.

The logger programs also produce a real-time hard-copy printout as an additional backup measure. This printout may be omitted if you wish, since the final log will be compiled and printed after the contest anyway. However, the hard copy is handy to have during the contest because it allows you to quickly obtain the "record number" needed to call up a back entry for review and editing. If it isn't too inconvenient, it's probably wise to keep a hard copy during the contest.

In addition to the running hard copy that is generated during the contest, the LOGPRINT program creates separate logs afterward. The post-contest logs differ in that they reflect all changes made with the EDIT feature. In addition, they can be organized on a band-by-band basis if you wish. Some operators will want to review the running log for errors after the contest and then use the "RESUME" feature to re-enter the logging program and edit out those errors. If you do that, however, be sure you don't use a real-time clock. All of the loggers except the FIELD DAY LOGGER are written to input automatically and save the times at which you log on and off. To avoid inadvertantly writing incorrect post-contest times into the log, use the manual time-keeping feature by answering "N" to the real-time clock question. Then you can prevent the saving of invalid times by simply typing a "0" in response to each instruction to enter the time.

Modifications for CP/M

To modify the TRS-80 listings for CP/M systems running Microsoft BASIC-80, make the following changes:

1. Delete the CLEAR command (line 35 in the GENERAL and FD loggers, line 40 in the SS LOGGER).

2. Use variables HI$ and LO$ (line 52 in the GENERAL and FD loggers, line 50 in SS) for your video highlighting commands (HI$ should be inverse, LO$ should be normal).

3. Delete all "CLS" commands and replace them with the clear-screen commands required for your system (in the GENERAL LOGGER, CLS is in lines 50, 122, 1380, 1700, 1770, 2310, 2750, 2790, 2950, and 3670; in the SS LOGGER, CLS is in lines 50, 152, 1390, 1630, 1700, 2270, 2720, 2760, 2900, and 3510; in the FD LOGGER, CLS is in lines 60, 122, 1290, 1610, 2000, 2420, 2460, 2630, and 3380).

4. The main title is displayed in double-size characters on a TRS-80. For 24 × 80 terminals, it should be centered with the TAB command and perhaps printed in inverse. The title is at lines 2310–2350 in the GENERAL LOGGER, lines 2000–2040 in the FD LOGGER, and lines 2270–2300 in the SS LOGGER.

5. The "IT'S-A-DUPE" message and the main menu use special TRS-80 graphics characters. The values 131 and 140 following the STRING$

command should both be replaced with 42 in the following lines: 600, 620, and 1400 on the GENERAL LOGGER, 650, 670, and 1310 in the FD LOGGER, and 590, 610, and 1410 in the SS LOGGER.

6. To prevent the menu from scrolling off the 16-line TRS-80 display, an extra INPUT statement appears in each logger. For 24 × 80 displays, this feature should be deleted entirely (delete lines 1515, 1517, and 1518 in the GENERAL LOGGER; lines 1425, 1427, and 1428 in the FD LOGGER; and lines 1525, 1526, and 1527 in the SS logger).

7. The time input subroutine in lines 125–128 of the GENERAL LOGGER is for a TRS-80 Model III only. It should be deleted for computers not having a real-time clock, and modified as necessary for systems requiring a different clock setup routine. In any case, the time-setting routine is there only as a convenience; the TRS-80, IBM PC, and other systems with internal clocks provide a direct means of setting the time and date, usually at initial sign-on time.

GENERAL CONTEST LOGGER (TRS-80 and CP/M)

```
10 REM GENERAL PURPOSE CONTEST LOGGER (TRS-80)
15 REM DESIGNED FOR DX/VHF CONTESTS, QSO PARTIES, ETC.
20 REM BY KC6A AND N6NB
30 REM COPYRIGHT 1983 BY HAYDEN BOOK CO.,INC.
35 CLEAR 1000
40 BL$=CHR$(7):REM DON'T DELETE--NEEDED.
50 CLS: REM CLEAR SCREEN
52 HI$="":LO$=""
70 FA$="DUPELOGA":FB$="DUPELOGB":FC$="CONTSTLG"
80 ET=0:ZA=0:AA=ASC("A")-1:A0=ASC("A")-ASC("0")+26
90 T1$="BAND      QSO'S X Q-PTS = POINTS  M'PLRS"
100 T2$="          -----          ------    ---"
105 DIM Y$(20),U1$(15),U2$(15),U3(15)
110 GOSUB 2310
120 PRINT:PRINT:A$="":INPUT "NEW CONTEST (N) OR RESUME (R) ";A$
122 CLS
123 INPUT "WANT TO USE THE REAL-TIME CLOCK TO LOG TIME";RC$
124 IF RC$<>"Y" THEN 130
125 INPUT "ENTER THE DATE (FORMAT:MM,DD,YY)";M5,D5,Y5
126 INPUT "ENTER THE TIME IN UTC (FORMAT: HH,MM)";HH,MM
127 POKE 16921,HH:POKE 16920,MM:REM TRS-80 MOD.III CLOCK INPUT
128 POKE 16924,M5:POKE 16923,D5:POKE 16922,Y5
130 IF A$="N" THEN  190
140 IF A$<>"R" THEN PRINT BL$:GOTO 110
150 GOSUB 2540:REM READ IN FILE
155 PRINT:PRINT "RESUMING ";Y$;" OF ";Z$
160 IF HC$="Y" THEN GOSUB 1070
170 SS=26*NL+1:GOTO 380
180 REM NEW CONTEST
190 PRINT:HC$="":AA$="":INPUT "WANT HARD COPY WHILE LOGGING (Y/N) ";HC$
200 IF HC$="Y" THEN INPUT "TURN ON PRINTER. <CR> WHEN READY ";AA$:GOSUB 1070
210 PRINT:Y$="":INPUT "WHAT IS THE CONTEST NAME ";Y$
215 PRINT:PRINT "NOTE: FOR CONTESTS IN WHICH YOU CANNOT WORK STATIONS"
216 PRINT "AGAIN ON ADDITIONAL BANDS, SPECIFY ONE LOG HERE AND USE THE"
217 PRINT "'MODE' SPECIFIER TO RECORD BAND CHANGES."
220 PRINT:INPUT "HOW MANY LOGS DO YOU NEED (USUALLY ONE PER BAND) ";NL:PRINT
230 IF NL<1 THEN PRINT BL$:GOTO 220
240 DIM LN$(NL),QP%(NL),TC%(NL),TS%(NL),SC%(100,NL)
250 M1=INT((FRE(0)-2000-800*NL)/6)
260 IF NL*26>=M1 THEN PRINT "TOO MANY";BL$:ERASE LN$,QP%,TC%,TS%,SC%:GOTO 220
270 DIM C1%(M1),C2%(M1),P%(M1),S$(100,NL)
275 KK$=""
280 FOR J=1 TO NL
```

```
290 PRINT:PRINT "WHAT BAND IS LOG SHEET #";J
300 LN$(J)="":INPUT " (E.G.80,20,144,1296 OR EVEN 10GHZ) ";LN$(J)
310 IF LEN(LN$(J))>8 THEN PRINT BL$;"NAME TOO LONG":GOTO 290
314 IF J=1 THEN 320
315 FOR K=1 TO J-1:IF LN$(J)=LN$(K) THEN PRINT "DUPLICATE BAND";BL$:GOTO 290
316 NEXT K
320 PRINT:INPUT "HOW MANY POINTS PER QSO ";QP%(J)
330 IF QP%(J)<1 THEN PRINT BL$:GOTO 320
331 KK$=KK$+LN$(J)+"  "
332 PRINT:PRINT "BANDS SPECIFIED: ";KK$:PRINT
340 TC%(J)=0:TS%(J)=0:NEXT J
350 T=1:L%=26*NL:SS=L%+1
352 P2=0:IF NL=1 THEN GOTO 355
353 PRINT:G$="":INPUT "ARE THERE SEPARATE MULTIPLIERS ON EACH BAND (Y/N) ";G$
354 IF G$="Y" THEN P2=1
355 PRINT:PRINT "YOU HAVE TWO OPTIONS FOR THE EXCHANGE: YOU CAN HAVE THE"
356 PRINT "ENTIRE EXCHANGE CONSTITUTE THE MULTIPLIER, OR DIVIDE OTHER"
357 PRINT "INFORMATION FROM THE MULTIPLIER WITH A SPACE (E.G. '599 ENY')"
359 PRINT:P1$="":INPUT "IS THE WHOLE EXCHANGE THE MULTIPLIER (Y/N) ";P1$
360 IF P1$="N" THEN PRINT "BE SURE TO PUT A SPACE BETWEEN THE REPORT AND MULTIPL
    IER."
365 PRINT:PRINT "SPACE IS AVAILABLE FOR APPROXIMATELY";M1;"CONTACTS.  GOOD LUCK!
    "
370 FOR J=1 TO L%:C1%(J)=-32000:P%(J)=0:NEXT J
380 GOSUB 3780
390 LG=1:LH%=0:LP=1:LL=0:SL$=""
400 GOSUB 1130:IF A$="N" THEN Z$=DTE$:REM DATE
410 IF HC$="Y" THEN GOSUB 1090
420 GOSUB 1590:REM BAND
430 GOSUB 1180:REM MODE
440 GOSUB 1380:REM MENU
442 ZB$="":PRINT:INPUT "<ENTER> TO LOG STARTING TIME";ZB$
443 ZA=ZA+1:IF RC$<>"Y" THEN 446
444 GOSUB 1162:U1$(ZA)=UTC$:GOTO 450
446 INPUT "ENTER THE TIME IN UTC (FORMAT: 0000)";U1$(ZA)
450 PRINT:PRINT "LOGGER READY FOR QSO #";T;"-";LN$(LG);"-";MD$
460 PRINT "COMMAND OR CALL:";
470 X1%=0:GOSUB 2370:IF X1%=1 THEN 450
480 IF G$="?" THEN GOSUB 1380:GOTO 450
490 IF G$="QSY" THEN GOSUB 1590:GOSUB 1700:GOTO 450
500 IF G$="DATE" THEN GOSUB 1130:GOTO 450
510 IF G$="MODE" THEN GOSUB 1180:GOTO 450
520 IF G$="QRT" THEN GOSUB 3900:GOSUB 3670:GOSUB 1860:GOTO 2000
525 IF G$="QRX" THEN GOSUB 3900:GOSUB 1870:CLOSE:GOSUB 3780:GOTO 442
530 IF G$="SAVE" THEN GOSUB 1870:CLOSE:GOSUB 3780:GOTO 450
535 IF G$="RATE" THEN GOSUB 3900:ET=E7:GOTO 450
540 IF G$="STAT" THEN GOSUB 1700:GOSUB 1780:GOTO 450
550 IF G$="SCORE" THEN GOSUB 3670:GOTO 450
560 IF G$="EDIT" THEN GOTO 2710
570 IF G$="X" THEN GOSUB 1230:GOTO 450
580 GOTO 690
590 PRINT HI$
600 PRINT STRING$(38,140)
610 PRINT "**  IT'S A DUPE  **  BOO  **  HISS  **"
620 PRINT STRING$(38,131):PRINT LO$;
630 GOTO 450
640 PRINT "INVALID ENTRY. PLEASE TRY AGAIN.";BL$:GOTO 450
650 PRINT "ABORTED":RETURN
660 PRINT "CAN'T DELETE";BL$:RETURN
670 PRINT "INVALID QSO #";BL$:GOTO 450
680 REM PROCESS A CALL SIGN
690 CS$=G$:CT$=G$
700 GOSUB 3400:ON X1% GOTO 590,820,640,710,710:REM DECODE CALL & JUMP ON RESULT
710 GOSUB 2400:IF LEN(G$)<>0 THEN 730
720 PRINT " -OK- ";
730 IF P1$="Y" THEN 740
732 SL$=G$:IF X1%=1 THEN 450
734 IF SL$="" THEN PRINT "REPORT (OR FULL EXCH)? ";:GOSUB 2370::GOTO 732
735 GOSUB 2400
740 SK$=G$:IF X1%=1 THEN 450
```

```
742 IF SK$="" THEN PRINT "MULTIPLIER? ";:GOSUB 2370:GOTO 740
750 IF LEN(SK$)+LEN(SL$)>14 THEN PRINT BL$;"EXCHANGE EXCEEDS 14 CHARACTERS":PRIN
    T:G$="":GOTO 730
752 IF LEN (SK$)<1 THEN PRINT BL$:PRINT:G$="":GOTO 740
760 IF RC$="Y" THEN GOSUB 1162:GOTO 770
762 GOSUB 2400:UTC$=G$:IF UTC$="" THEN PRINT "UTC? ";:GOSUB 2370:UTC$=G$:IF X1%=
    1 THEN 450
770 RN=T:GOSUB 900:REM WRITE LOG TO DISK
780 IF HC$="Y" THEN GOSUB 960:REM WRITE TO PRINTER
790 LH%=H%:GOSUB 990:REM ENTER IN DUPE SHEET
800 GOSUB 1020:REM ENTER SECTION
810 TC%(LG)=TC%(LG)+1:T=T+1:GOTO 450
820 PRINT:PRINT TAB(2)CT$;" NEEDED ON:"
830 FOR J=1 TO NL
840 H%=E(4)+J*26-26:GOSUB 3610:REM DUPE SEARCH
850 IF X1%=1 THEN 880
860 IF J=LG THEN PRINT TAB(10)"* ";ELSE PRINT "              ";
870 PRINT LN$(J)
880 NEXT J:PRINT:GOTO 450
890 REM WRITE LOG TO DISK
900 PRINT "...LOGGING"
910 LSET D1$=DTE$:LSET TM$=UTC$:LSET KL$=CT$
920 LSET B1$=LN$(LG):LSET M1$=MD$:LSET E1$=SL$+" "+SK$
930 PUT 3,RN
940 RETURN
950 REM WRITE LOG TO PRINTER
960 LPRINT T;TAB(6)DTE$;TAB(17)UTC$;TAB(24)LN$(LG);TAB(30)MD$;TAB(36)CT$;TAB(48)
    SL$;" ";SK$
970 RETURN
980 REM ENTER IN DUPE LIST
990 IF X1%=5 THEN L%=L%+1:P%(H%)=L%:H%=L%
1000 C1%(H%)=E1%:C2%(H%)=E2%:P%(H%)=0:RETURN
1010 REM UPDATE SECTION
1020 IF P2=0 THEN LF=1 ELSE LF=LG
1022 FOR J=1 TO TS%(LF):IF SK$=S$(J,LF) THEN 1060
1030 NEXT J
1040 TS%(LF)=TS%(LF)+1:J=TS%(LF):SC%(J,LF)=0:S$(J,LF)=SK$
1050 PRINT " * NEW MULTIPLIER *"
1060 SC%(J,LF)=SC%(J,LF)+1:RETURN
1070 REM PRINTER INIT, IF USED
1080 RETURN
1090 LPRINT:LPRINT:LPRINT TAB(5)"CONTEST: ";Y$
1100 LPRINT TAB(5)"DATE:    ";Z$
1110 LPRINT:RETURN
1120 REM SET DATE
1130 IF RC$="Y" THEN 1162
1132 GOSUB 2400
1140 IF G$="" THEN PRINT:INPUT "WHAT IS THE DATE (MM/DD/YY) ";G$
1150 IF LEN(G$)>8 THEN PRINT "DATE TOO LONG";BL$:G$="":GOTO 1140
1160 DTE$=G$:RETURN
1162 DTE$=LEFT$(TIME$,8):UTC$=MID$(TIME$,10,2)+MID$(TIME$,13,2)
1164 PRINT "UTC=";UTC$:RETURN
1170 REM SET MODE
1180 GOSUB 2400
1190 IF G$="" THEN PRINT:INPUT "WHAT IS THE MODE ";G$
1200 IF LEN(G$)>3 THEN PRINT "INVALID MODE. 3 CHARACTERS MAX.";BL$:G$="":GOTO 11
    90
1210 MD$=G$:RETURN
1220 REM X DELETES LAST ENTRY
1230 IF LH%=0 THEN 660
1240 IF LH%<SS THEN IF P%(LH%)=0 THEN C1%(LH%)=-32000:GOTO 1260
1250 P%(LH%)=0:L%=L%-1
1260 T=T-1:TC%(LG)=TC%(LG)-1:LH%=0
1270 GOSUB 1310
1280 IF HC$="Y" THEN LPRINT "-- PREVIOUS CALL DELETED --"
1290 PRINT " DELETED":RETURN
1300 REM DELETE SECTION
1310 IF P2=0 THEN LF=1 ELSE LF=LG
1312 FOR J=1 TO TS%(LF):IF SK$=S$(J,LF) THEN 1330
1320 NEXT J
```

```
1330 SC%(J,LF)=SC%(J,LF)-1:IF SC%(J,LF)>0 THEN RETURN
1340 IF J=TS%(LF) THEN 1360
1350 SC%(J,LF)=SC%(TS%(LF),LF):S$(J,LF)=S$(TS%(LF),LF)
1360 TS%(LF)=TS%(LF)-1:RETURN
1370 REM MENU
1380 CLS:PRINT TAB(4)STRING$(27,140)
1390 PRINT HI$;"     GENERAL CONTEST LOGGER MENU          "
1400 PRINT TAB(4)STRING$(27,131)
1410 PRINT " '?' DISPLAYS THIS MENU                       "
1420 PRINT " 'QSY <BAND> <MODE>' CHANGES BANDS            "
1430 PRINT " 'DATE <MM/DD/YY>' CHANGES DATE               "
1440 PRINT " 'MODE <MODE>' CHANGES MODES                  "
1450 PRINT " 'QRT' SAVES ALL CALLS AND EXITS              "
1455 PRINT " 'QRX' SAVES ALL CALLS AND STOPS TIMER        "
1460 PRINT " 'SAVE' SAVES ALL CALLS AND RESUMES           "
1470 PRINT "         (NOTE: ALLOW 2-3 MIN FOR SAVE)       "
1475 PRINT " 'RATE' SHOWS TOTAL TIME USED AND RATE        "
1480 PRINT " 'STAT' GIVES STATUS ON CURRENT BAND          "
1490 PRINT " 'SCORE' GIVES RUNNING TOTAL SCORE            "
1495 CR$="":INPUT "<ENTER> TO CONTINUE";CR$
1497 PRINT TAB(10) "MENU (CONTINUED)"
1498 PRINT TAB(10)STRING$(16,131)
1500 PRINT " 'EDIT <QSO #>' EDITS ANY LOGGED ENTRY        "
1510 PRINT " 'X' DELETES THE LAST ENTRY                   "
1520 PRINT " '-' ABORTS CURRENT ENTRY                     "
1530 PRINT " ENTRY FORMAT: 'CALL EXCHANGE <UTC>'          "
1540 PRINT " ENTER THE CALL ONLY FOR DUPE CHECK           "
1550 PRINT " ENTER CALL AND '?' TO LIST BANDS             "
1552 PRINT "       ON WHICH A STATION IS NEEDED           "
1555 PRINT "                                              "
1560 PRINT " <> DENOTES OPTIONAL ENTRIES IN MENU          "
1570 PRINT LO$:RETURN
1580 REM QSY
1590 GOSUB 2400
1600 IF NL=1 THEN RETURN
1610 IF G$="" THEN PRINT:INPUT "WHAT BAND (LOG SHEET) DO YOU WANT ";G$
1620 FOR J=1 TO NL: IF G$=LN$(J) THEN 1660
1630 NEXT J
1640 PRINT "NO SUCH LOG. VALID LOGS ARE:";BL$
1650 FOR J=1 TO NL:PRINT TAB(5)LN$(J):NEXT J:G$="":GOTO 1610
1660 LG=J:IF HC$="Y" THEN GOSUB 2490
1670 GOSUB 2400:IF G$<>"" THEN GOSUB 1200
1680 RETURN
1690 REM STATUS
1700 CLS:PRINT:PRINT "DATE= ";DTE$;
1710 PRINT "  BAND= ";LN$(LG);
1720 PRINT "  MODE= ";MD$
1725 IF P2=0 THEN LF=1 ELSE LF=LG
1730 PRINT " SECTIONS WORKED= ";TS%(LF);
1740 PRINT "  CONTACTS= ";TC%(LG)
1750 PRINT:RETURN
1760 REM LIST SECTIONS
1770 CLS:PRINT:IF P2=0 THEN LF=1 ELSE LF=LG
1780 PRINT "------ MULTIPLIERS WORKED";:IF P2=1 THEN PRINT " ON ";LN$(LF);
1782 PRINT " ------"
1790 IF TS%(LF)=0 THEN RETURN
1800 TB=1
1810 FOR J=1 TO TS%(LF)
1820 PRINT TAB(TB)S$(J,LF);
1830 TB=TB+8:IF TB>60 THEN TB=1:PRINT
1840 NEXT J:PRINT:RETURN
1850 REM QRT
1860 CLOSE 3
1870 OPEN "O",1,FA$
1880 OPEN "R",2,FB$,6
1890 FIELD 2,2 AS C1$,2 AS C2$,2 AS P$
1900 PRINT#1,HC$:PRINT#1,T:PRINT#1,Y$:PRINT#1,Z$:PRINT#1,NL:PRINT#1,L%:PRINT#1,M
     1
1910 FOR J=1 TO NL:PRINT#1,LN$(J):NEXT J
1912 PRINT#1,ZA:PRINT#1,ET:FOR J=1 TO ZA:PRINT#1,U1$(J):PRINT#1,U2$(J):PRINT#1,U
     3(J):NEXT J
```

```
1920 FOR J=1 TO L%
1930 LSET C1$=MKI$(C1%(J)):LSET C2$=MKI$(C2%(J)):LSET P$=MKI$(P%(J))
1940 PUT 2,J:NEXT J:CLOSE 2
1950 FOR J=1 TO NL:PRINT#1,QP%(J);",";TC%(J);",";TS%(J):NEXT J
1960 FOR J=1 TO NL:FOR K=1 TO TS%(J)
1970 PRINT#1,SC%(K,J);",";S$(K,J)
1980 NEXT K:NEXT J
1982 PRINT#1,P1$:PRINT#1,P2
1985 RETURN
1990 REM WRITE SECT SUMMARY TO #1
2000 PRINT#1,BL$:REM NEW PAGE
2010 PRINT#1,TAB(27)"* MULTIPLIER SUMMARY *"
2015 IF P2=0  THEN LP=1 ELSE LP=NL
2020 FOR J=1 TO LP
2030 PRINT#1," ":PRINT#1," ":PRINT#1," "
2040 PRINT#1,STRING$(21,45);"  MULTIPLIERS";:IF P2=1 THEN PRINT#1," WORKED ON ";
     LN$(J);
2042 PRINT#1,"  ";STRING$(21,45):PRINT#1," "
2050 FOR K=1 TO 5:PRINT#1,"  M'PLIER/QSOS ";:NEXT K:PRINT#1
2060 TB=9
2070 FOR K=1 TO TS%(J)
2080 PRINT#1,TAB(TB-LEN(S$(K,J)))S$(K,J);"/";SC%(K,J);
2090 TB=TB+15:IF TB>74 THEN TB=9:PRINT#1,
2100 NEXT K:PRINT#1," ":NEXT J
2110 REM WRITE SCORE TO LOG#1
2120 PRINT #1,BL$:REM NEW PAGE
2130 PRINT#1,TAB(14)"SCORE SUMMARY":PRINT#1,:PRINT#1,T1$
2140 IF HC$="Y" THEN LPRINT:LPRINT TAB(14)"SCORE SUMMARY":LPRINT:LPRINT T1$
2150 X2=0:X3=0:X4=0
2160 FOR J=1 TO NL
2170 X4=X4+TC%(J):X1=TC%(J)*QP%(J):X2=X2+X1:X3=X3+TS%(J)
2180 PRINT#1,LN$(J);TAB(11)TC%(J);TAB(20)QP%(J);TAB(27)X1;:IF P2=1 THEN PRINT#1,
     TAB(37)TS%(J);
2182 PRINT#1,
2188 IF HC$<>"Y" THEN 2200
2190 LPRINT LN$(J);TAB(11)TC%(J);TAB(20)QP%(J);TAB(27)X1;:IF P2=1 THEN LPRINT TA
     B(37)TS%(J);
2192 LPRINT
2200 NEXT J
2208 IF P2=0 THEN X3=TS%(1)
2210 PRINT#1,T2$:PRINT#1,"TOTAL";TAB(11)X4;TAB(27)X2;TAB(37)X3:PRINT#1,
2220 IF HC$="Y" THEN LPRINT T2$:LPRINT "TOTAL";TAB(11)X4;TAB(27)X2;TAB(37)X3:LPR
     INT
2230 PRINT#1,TAB(25)"***********"
2240 PRINT#1,TAB(25)"*         *"
2250 PRINT#1,"SCORE = ";X2;" X ";X3;TAB(25)"* ";X2*X3;TAB(35)"*"
2260 PRINT#1,TAB(25)"*         *"
2270 PRINT#1,TAB(25)"***********"
2280 CLOSE
2290 IF HC$="Y" THEN LPRINT "SCORE = ";X2*X3
2300 END
2310 CLS:PRINT:REM PRINT TITLE
2320 PRINT CHR$(23):PRINT
2330 PRINT "GENERAL CONTEST LOGGING PROGRAM"
2335 PRINT:PRINT TAB(8)"BY KC6A AND N6NB"
2340 PRINT "(C) 1983 BY HAYDEN BOOK CO.,INC."
2350 PRINT:PRINT:RETURN
2360 REM INPUT SCAN ROUTINE
2370 GS$="":LINE INPUT GS$:LL=LEN(GS$)
2380 IF INSTR(GS$,"-")>0 THEN X1%=1:GOTO 650
2390 LP=1
2400 IF LP>LL THEN 2430:REM 2ND ENTRY
2410 FOR LT=LP TO LL:IF MID$(GS$,LT,1)<>" " THEN 2440
2420 NEXT LT
2430 G$="":RETURN
2440 LP=LT
2450 FOR LT=LP TO LL:IF MID$(GS$,LT,1)=" " THEN 2480
2460 NEXT LT
2470 LT=LT+1
2480 G$=MID$(GS$,LP,LT-LP):LP=LT:RETURN
2490 LPRINT
```

```
2500 LPRINT TAB(5)"QSY TO: ";LN$(LG)
2510 RETURN
2520 LL=LEN(GS$):GOTO 2390
2530 REM RESUME
2540 OPEN "I",1,FA$
2550 OPEN "R",2,FB$,6
2560 FIELD 2,2 AS C1$,2 AS C2$,2 AS P$
2570 INPUT#1,HC$:INPUT#1,T:INPUT#1,Y$:INPUT#1,Z$:INPUT#1,NL:INPUT#1,L%:INPUT#1,M
     1
2580 DIM LN$(NL),QP%(NL),TC%(NL),TS%(NL),SC%(100,NL)
2590 DIM C1%(M1),C2%(M1),P%(M1),S$(100,NL)
2600 FOR J=1 TO NL:INPUT#1,LN$(J):NEXT J
2602 INPUT#1,ZA:INPUT#1,ET:FOR J=1 TO ZA:INPUT#1,U1$(J):INPUT#1,U2$(J):INPUT#1,U
     3(J):NEXT J
2610 FOR J=1 TO L%
2620 GET 2,J
2630 C1%(J)=CVI(C1$):C2%(J)=CVI(C2$):P%(J)=CVI(P$)
2640 NEXT J
2650 FOR J=1 TO NL:INPUT#1,QP%(J),TC%(J),TS%(J):NEXT J
2660 FOR J=1 TO NL:FOR K=1 TO TS%(J)
2670 INPUT#1,SC%(K,J),S$(K,J)
2680 NEXT K:NEXT J
2682 INPUT#1,P1$:INPUT#1,P2
2690 CLOSE:RETURN
2700 REM EDIT A RECORD
2710 GOSUB 2400:IF G$="" THEN INPUT "WHAT QSO# DO YOU WANT TO EDIT ";G$
2720 X2=INT(VAL(G$))
2730 IF X2>0 THEN IF X2<T THEN 2750
2740 GOTO 670
2750 CLS:IF P1$="Y" THEN JB=7 ELSE JB=8
2760 GET 3,X2:Y$(2)=D1$:Y$(3)=TM$:Y$(4)=KL$:Y$(5)=B1$:Y$(6)=M1$
2764 FOR J=2 TO 6:GS$=Y$(J):GOSUB 2520:Y$(J)=G$:Y$(J+10)=G$:NEXT J
2765 GS$=E1$:GOSUB 2520:Y$(7)=G$:Y$(17)=G$
2766 Y$(8)=Y$(7):IF P1$<>"Y" THEN GOSUB 2400:Y$(8)=G$
2767 Y$(18)=Y$(8)
2770 GOSUB 3080
2780 PRINT:A$="":INPUT "CHANGE THIS RECORD (Y/N) ";A$
2790 IF A$<>"Y" THEN CLS:GOTO 450
2800 LH%=0
2810 FOR J=1 TO NL:IF Y$(5)=LN$(J) THEN 2830
2820 NEXT J
2830 LX=LG:LY=J:LK=LY
2840 PRINT:PRINT TAB(12)"INSTRUCTIONS":PRINT
2850 PRINT "HIT <CR> TO LEAVE AN ITEM UNCHANGED."
2860 PRINT "TO CHANGE, ENTER NEW VALUE <CR>."
2870 PRINT "TO DELETE THIS ENTRY, CHANGE CALL TO '*'."
2880 PRINT:PRINT TAB(5)"DATE:";TAB(14)Y$(2);:A$="":INPUT A$:IF A$<>"" THEN Y$(2)
     =A$
2890 PRINT TAB(5)"TIME:";TAB(14)Y$(3);:A$="":INPUT A$:IF A$<>""THEN Y$(3)=A$
2900 PRINT TAB(5)"CALL:";TAB(14)Y$(4);:A$="":INPUT A$:IF A$<>""THEN Y$(4)=A$
2910 PRINT TAB(5)"BAND:";TAB(14)Y$(5);:A$="":INPUT A$:IF A$<>"" THEN Y$(5)=A$
2920 PRINT TAB(5)"MODE:";TAB(14)Y$(6);:A$="":INPUT A$:IF A$<>""THEN Y$(6)=A$
2930 IF P1$<>"Y" THEN PRINT TAB(5)"EXCHANGE:";TAB(14)Y$(7);:A$="":INPUT A$:IF A$
     <>"" THEN Y$(7)=A$
2935 PRINT TAB(5)"M'PLIER:";TAB(14)Y$(8);:INPUT A$:IF A$<>"" THEN Y$(8)=A$
2940 PRINT:A$="":INPUT "OK (Y/N) ";A$:X1%=4
2950 IF A$<>"Y" THEN CLS:GOSUB 650:GOTO 450
2960 IF Y$(5)<>Y$(15) THEN GOSUB 3240:GOTO 3000:REM BAND CHANGE
2970 LG=LK
2980 IF Y$(4)<>Y$(14) THEN GOSUB 3160:GOSUB 3190:REM CALL CHANGE
2990 IF Y$(8)<>Y$(18) THEN GOSUB 3320:GOSUB 3350:REM EXCHANGE CHANGE
3000 LG=LX
3010 ON X1% GOTO 590,3370,3370,3020,3020,3380
3020 LSET D1$=Y$(2):LSET TM$=Y$(3):LSET KL$=Y$(4)
3030 LSET B1$=Y$(5):LSET M1$=Y$(6)
3032 IF P1$="Y" THEN Y$(7)=""
3034 LSET E1$=Y$(7)+" "+Y$(8)
3040 PUT 3,X2
3050 IF HC$<>"Y" THEN 3070
3060 LPRINT "  ** EDIT ON REC #";X2;": ";D1$;" ";TM$;" ";KL$;" ";B1$;" ";M1$;" "
     ;E1$
```

```
3070 GOTO 450
3080 PRINT:PRINT  TAB(5)"DATE:";TAB(14)Y$(2)
3090 PRINT TAB(5)"TIME:";TAB(14)Y$(3)
3100 PRINT TAB(5)"CALL:";TAB(14)Y$(4)
3110 PRINT TAB(5)"BAND:";TAB(14)Y$(5)
3120 PRINT TAB(5)"MODE:";TAB(14)Y$(6)
3130 IF P1$<>"Y" THEN PRINT TAB(5)"EXCHANGE:";TAB(14)Y$(7)
3135 PRINT TAB(5)"M'PLIER:";TAB(14)Y$(8)
3140 RETURN
3150 REM CALL CHANGED
3160 IF Y$(4)="*" THEN Y$(8)="*":RETURN
3170 CS$=Y$(4):GOSUB 3400:IF X1%<4 THEN RETURN:REM CHECK IF NEW CALL IS DUPE
3180 GOSUB 990:TC%(LG)=TC%(LG)+1:RETURN
3190 IF X1%<4 THEN RETURN
3195 IF Y$(14)="*" THEN RETURN
3200 CS$=Y$(14)
3210 GOSUB 3400:C1%(H%)=-32001:C2%(H%)=-32000:REM DELETE OLD CALL
3220 TC%(LG)=TC%(LG)-1:X1%=4:RETURN
3230 REM CHANGE BAND
3240 FOR LK=1 TO NL:IF Y$(5)=LN$(LK) THEN 3270
3250 NEXT LK
3260 X1%=6:RETURN
3270 LG=LK:GOSUB 3160:IF X1%<4 THEN RETURN
3280 LG=LY:GOSUB 3190
3290 GOSUB 3320:LG=LK:GOSUB 3350:RETURN
3300 LG=LY:GOSUB 3350:RETURN
3310 REM CHANGE EXCH
3320 IF Y$(18)="*" THEN RETURN
3330 SK$=Y$(18)
3340 GOTO 1310:REM DELETE OLD SECTION
3350 IF Y$(8)="*" THEN RETURN
3360 SK$=Y$(8):GOSUB 1020:RETURN:REM ADD NEW SECTION
3370 PRINT BL$;"INVALID CALL SIGN":GOTO 450
3380 PRINT BL$;"INVALID BAND":GOTO 450
3390 REM DUPE CHECKER: X1%=1=DUPE: 2=QUERY: 3=INVALID CALL: 4,5=OK
3400 X1%=4:E(5)=0:E(6)=0:K=1:IF LEN(CS$)>10 THEN X1%=3:RETURN
3410 IF RIGHT$(CS$,1)="?" THEN CS$=LEFT$(CS$,LEN(CS$)-1):X1%=2:CT$=CS$
3420 IF LEN(CS$)<4 THEN X1%=3:RETURN
3430 IF MID$(CS$,3,1)<"A" THEN 3460
3440 IF MID$(CS$,2,1)=>"A" THEN X1%=3:RETURN
3450 CS$=" "+CS$:K=2:E(1)=0
3460 FOR J=K TO LEN(CS$)
3470 W$=MID$(CS$,J,1):IF W$="/" THEN 3520
3480 E(J)=ASC(W$)-AA:IF E(J)>26 THEN X1%=3:RETURN
3490 IF W$<"0" THEN X1%=3:RETURN
3500 IF E(J)<1 THEN E(J)=E(J)+A0:IF E(J)>36 THEN X1%=3:RETURN
3510 NEXT J
3520 IF J>7 THEN X1%=3:RETURN
3530 IF J-K<4 THEN X1%=3:RETURN
3540 FOR J=4 TO 6:IF E(J)>26 THEN X1%=3:RETURN
3550 NEXT J
3560 E1%=-32000+E(1)*1369+E(2)*37+E(3)
3570 E2%=-32000+E(4)*1369+E(5)*37+E(6)
3580 IF X1%=2 THEN RETURN
3590 H%=E(4)+LG*26-26
3600 REM SEARCH DUPE STRING
3610 IF C1%(H%)=-32000 THEN X1%=4:RETURN
3620 IF C2%(H%)<>E2% THEN 3640
3630 IF C1%(H%)=E1% THEN X1%=1:RETURN
3640 IF P%(H%)=0 THEN X1%=5:RETURN
3650 H%=P%(H%):GOTO 3620
3660 REM SCORE
3670 PRINT:PRINT TAB(22)"SCORE SUMMARY":PRINT
3680 PRINT T1$
3690 X2=0:X3=0:X4=0
3700 FOR J=1 TO NL
3710 X4=X4+TC%(J):X1=TC%(J)*QP%(J):X2=X2+X1:X3=X3+TS%(J)
3720 PRINT LN$(J);TAB(11)TC%(J);TAB(20)QP%(J);TAB(27)X1;
3722 IF P2=1 THEN PRINT TAB(37)TS%(J);
```

```
3724 PRINT
3730 NEXT J
3740 PRINT T2$
3750 PRINT "TOTAL";TAB(11)X4;TAB(27)X2;TAB(37)X3
3760 PRINT:PRINT TAB(10)"SCORE =";X3*X2
3770 PRINT:RETURN
3780 OPEN "R",3,FC$,47
3790 FIELD 3,8 AS D1$,4 AS TM$,10 AS KL$,5 AS B1$,5 AS M1$,15 AS E1$
3800 RETURN
3900 IF RC$<>"Y" THEN 3920
3910 GOSUB 1162:U2$(ZA)=UTC$:GOTO 3930
3920 PRINT:INPUT "ENTER THE TIME IN UTC";U2$(ZA)
3930 UX=VAL(U1$(ZA)):UY=VAL(U2$(ZA))
3932 U7=INT(UX/100):U4=((UX/100)-U7)/.6:U7=U7+U4
3940 U5=INT(UY/100):U6=((UY/100)-U5)/.6:U5=U5+U6
3950 IF U5<U7 THEN U5=U5+24
3960 E7=ET:ET=ET+U5-U7:U3(ZA)=T:IF ET=0 THEN 3990
3970 PRINT:PRINT "TIME USED SO FAR:";INT(ET);" HRS - ";INT((ET-INT(ET))*60+.5);"
     MINS"
3980 PRINT:PRINT "OVERALL RATE: ";INT ((T-1)/ET+.5);" QSOS PER HOUR"
3990 RETURN
```

GENERAL CONTEST LOGGER (Apple)

```
10  REM  GENERAL PURPOSE CONTEST LOGGER (APPLE)
15  REM  DESIGNED FOR DX/VHF CONTESTS, QSO PARTIES, ETC.
20  REM  BY KC6A AND N6NB
30  REM  (C) 1983 BY HAYDEN BOOK CO.,INC.
40 BL$ =  CHR$ (7): REM  BELL
50 D$ =  CHR$ (4):Q$ =  CHR$ (34)
52 LP$ = D$ + "PR#1":SCN$ = D$ + "PR#0"
54 RD$ = D$ + "READ ":WR$ = D$ + "WRITE "
56 CL$ = D$ + "CLOSE ":OP$ = D$ + "OPEN "
57 DL$ = D$ + "DELETE "
58 DV$ = ",D1"
60  HOME
70 F1$ = "DUPELOGA"
72 F2$ = "DUPELOGB":R2$ = ",L19"
74 F3$ = "CONTSTLG":R3$ = ",L58"
80 AA =  ASC ("A") - 1:A0 =  ASC ("A") -  ASC ("0") + 26:ET = 0:ZA = 0
90 T1$ = "BAND       QSO'S X Q-PTS = POINTS M'PLRS"
100 T2$ = "          -----          ------    ---"
105  DIM Y$(20),U1$(15),U2$(15),U3(15)
110  GOSUB 2310
115  PRINT : PRINT "ARE YOU USING A REAL TIME CLOCK FOR"
116  INPUT "LOGGING THE TIME (Y/N)? ";RC$
120  PRINT : INPUT "NEW CONTEST (N) OR RESUME (R)? ";A$
130  IF A$ = "N" THEN 190
140  IF A$ <  > "R" THEN  PRINT BL$: GOTO 110
150  GOSUB 2540: REM  READ IN FILE
155  PRINT "RESUMING ";Y$;" OF ";Z$
160  IF HC$ = "Y" THEN  GOSUB 1070
170 SS = 26 * NL + 1: GOTO 390
180  REM  NEW CONTEST
190  PRINT : INPUT "WANT HARD COPY WHILE LOGGING (Y/N)? ";HC$
200  IF HC$ = "Y" THEN  INPUT "TURN ON PRINTER. <CR> WHEN READY? ";AA$: GOSUB 1070
210  PRINT : INPUT "WHAT IS THE CONTEST NAME? ";Y$
212  PRINT : PRINT "NOTE:  FOR CONTESTS IN WHICH YOU"
213  PRINT "CANNOT WORK STATIONS AGAIN FOR CREDIT"
214  PRINT "ON ADDITIONAL BANDS, SPECIFY ONE LOG"
215  PRINT "HERE AND USE THE 'MODE' TO RECORD"
216  PRINT "BAND CHANGES."
```

```
220  PRINT : PRINT "HOW MANY LOGS DO YOU NEED": INPUT "(USUALLY ONE PER BAND)? ";N
L: PRINT
230  IF NL < 1 THEN  PRINT BL$: GOTO 220
240  DIM LN$(NL),QP%(NL),TC%(NL),TS%(NL),SC%(100,NL)
242 M1 =  FRE (0): IF M1 < 0 THEN M1 = M1 + 65536
250 M1 =  INT ((M1 - 2000 - 800 * NL) / 6)
260  IF NL * 26 >  = M1 THEN  PRINT "TOO MANY";BL$: END
270  DIM C1%(M1),C2%(M1),P%(M1),S$(100,NL)
275 KK$ = ""
280  FOR J = 1 TO NL
290  PRINT : PRINT "WHAT BAND IS LOG SHEET #";J
300  INPUT " (E.G. 80,20,144,1296 OR EVEN 10GHZ)? ";LN$(J)
310  IF  LEN (LN$(J)) > 8 THEN  PRINT BL$;"NAME TOO LONG": GOTO 290
314  IF J = 1 THEN 320
315  FOR K = 1 TO J - 1: IF LN$(J) = LN$(K) THEN  PRINT "DUPLICATE BAND";BL$: GOTO
 290
316  NEXT K
320  PRINT : INPUT "HOW MANY POINTS PER QSO? ";QP%(J)
330  IF QP%(J) < 1 THEN  PRINT BL$: GOTO 320
331 KK$ = KK$ + LN$(J) + "  "
332  PRINT : PRINT "BANDS SPECIFIED: ";KK$: PRINT
340 TC%(J) = 0:TS%(J) = 0: NEXT J
350 T = 1:L% = 26 * NL:SS = L% + 1
352 P2 = 0: IF NL = 1 THEN  GOTO 355
353  PRINT : PRINT "ARE THERE SEPARATE MULTIPLIERS": INPUT "ON EACH BAND (Y/N)?  "
;G$
354  IF G$ = "Y" THEN P2 = 1
355  PRINT : PRINT "YOU HAVE TWO OPTIONS FOR THE EXCHANGE:"
356  PRINT "YOU CAN TREAT THE ENTIRE EXCHANGE AS A"
357  PRINT "MULTIPLIER, OR DIVIDE OTHER INFORMATION"
358  PRINT "FROM THE MULTIPLIER WITH A SPACE,"
359  PRINT "AS IN '599 ENY'."
360  PRINT "IS THE WHOLE EXCHANGE THE MULTIPLIER?"
361  INPUT "ENTER Y/N: ";P1$
362  IF P1$ = "N" THEN  PRINT "DON'T FORGET THE SPACE BETWEEN"
363  IF P1$ = "N" THEN  PRINT "THE REPORT AND MULTIPLIER!"
365  PRINT : PRINT "SPACE IS AVAILABLE FOR APPROXIMATELY": PRINT M1;" CONTACTS.  G
OOD LUCK!"
370  FOR J = 1 TO L%:C1%(J) =  - 32000:P%(J) = 0: NEXT J
390 LG = 1:LH% = 0:LP = 1:LL = 0:SL$ = ""
400  GOSUB 1130: IF A$ = "N" THEN Z$ = DTE$: REM  DATE
410  IF HC$ = "Y" THEN  GOSUB 1090
420  GOSUB 1590: REM  BAND
430  GOSUB 1180: REM  MODE
440  GOSUB 1380: REM  MENU
442  PRINT : INPUT "<CR> TO LOG STARTING TIME";ZB$
443 ZA = ZA + 1: IF RC$ <  > "Y" THEN 446
444  GOSUB 1162:U1$(ZA) = UTC$: GOTO 450
446  INPUT "ENTER THE TIME IN UTC";U1$(ZA)
450  PRINT : PRINT "LOGGER READY FOR QSO #";T;"-";LN$(LG);"-";MD$
460  PRINT "COMMAND OR CALL: ";
470 X1% = 0: GOSUB 2370: IF X1% = 1 THEN 450
480  IF G$ = "?" THEN  GOSUB 1380: GOTO 450
490  IF G$ = "QSY" THEN  GOSUB 1590: GOSUB 1700: GOTO 450
500  IF G$ = "DATE" THEN  GOSUB 1130: GOTO 450
510  IF G$ = "MODE" THEN  GOSUB 1180: GOTO 450
520  IF G$ = "QRT" THEN  GOSUB 3920: GOSUB 3660: GOSUB 1860: GOSUB 2000: GOTO 2290
525  IF G$ = "QRX" THEN  GOSUB 3920: GOSUB 1860: GOSUB 2280: GOTO 442
530  IF G$ = "SAVE" THEN  GOSUB 1860: GOSUB 2280: GOTO 450
535  IF G$ = "RATE" THEN  GOSUB 3920:ET = E7: GOTO 450
540  IF G$ = "STAT" THEN  GOSUB 1700: GOSUB 1780: GOTO 450
550  IF G$ = "SCORE" THEN  GOSUB 3660: GOTO 450
560  IF G$ = "EDIT" THEN  GOTO 2710
570  IF G$ = "X" THEN  GOSUB 1230: GOTO 450
580  GOTO 690
590  PRINT BL$: FLASH
600  PRINT "****************************************"
610  PRINT "**  IT'S A DUPE  **  BOO  **  HISS  **"
620  PRINT "****************************************"
630  NORMAL : GOTO 450
```

```
640  PRINT "INVALID ENTRY. PLEASE TRY AGAIN.";BL$: GOTO 450
650  PRINT "ABORTED": RETURN
660  PRINT "CAN'T DELETE";BL$: RETURN
670  PRINT "INVALID QSO #";BL$: GOTO 450
680  REM  PROCESS A CALL SIGN
690 CS$ = G$:CT$ = G$
700  GOSUB 3400: ON X1% GOTO 590,820,640,710,710: REM  DECODE CALL & JUMP ON RESUL
T
710  GOSUB 2400: IF  LEN (G$) <  > 0 THEN 730
720  PRINT " -OK- ";
730  IF P1$ = "Y" THEN 740
732 SL$ = G$: IF X1% = 1 THEN 450
734  IF SL$ = "" THEN  PRINT "REPORT (OR FULL EXCH) ";: GOSUB 2370: GOTO 732
735  GOSUB 2400
740 SK$ = G$: IF X1% = 1 THEN 450
742  IF SK$ = "" THEN  PRINT "MULTIPLIER ";: GOSUB 2370: GOTO 740
750  IF  LEN (SK$) +  LEN (SL$) > 14 THEN  PRINT BL$;"EXCHANGE EXCEEDS 14 CHARACTE
RS": PRINT :G$ = "": GOTO 730
752  IF  LEN (SK$) < 1 THEN  PRINT BL$: PRINT :G$ = "": GOTO 740
760  IF RC$ = "Y" THEN  GOSUB 1162: GOTO 770
762  GOSUB 2400:UTC$ = G$: IF UTC$ = "" THEN  PRINT "UTC ";: GOSUB 2370:UTC$ = G$:
 IF X1% = 1 THEN 450
770 RN = T: GOSUB 900: REM  WRITE LOG TO DISK
780  IF HC$ = "Y" THEN  GOSUB 960: REM  WRITE TO PRINTER
790 LH% = H%: GOSUB 990: REM  ENTER IN DUPE SHEET
800  GOSUB 1020: REM  ENTER SECTION
810 TC%(LG) = TC%(LG) + 1:T = T + 1: GOTO 450
820  PRINT : PRINT  TAB( 2)CT$;" NEEDED ON:"
830  FOR J = 1 TO NL
840 H% = E(4) + J * 26 - 26: GOSUB 3610: REM  DUPE SEARCH
850  IF X1% = 1 THEN 880
860  PRINT  TAB( 11);: IF J = LG THEN  FLASH
870  PRINT LN$(J): NORMAL
880  NEXT J: PRINT : GOTO 450
890  REM  WRITE LOG TO DISK
900  PRINT "..LOGGING"
910  PRINT OP$;F3$;R3$;DV$
912  PRINT WR$;F3$;",R";RN
920  GOSUB 3900
930  PRINT CL$
940  RETURN
950  REM  WRITE LOG TO PRINTER
960  PRINT LP$: PRINT T; SPC( 5 -  LEN ( STR$ (T)));: GOSUB 3900: PRINT SCN$.
970  RETURN
980  REM  ENTER IN DUPE LIST
990  IF X1% = 5 THEN L% = L% + 1:P%(H%) = L%:H% = L%
1000 C1%(H%) = E1%:C2%(H%) = E2%:P%(H%) = 0: RETURN
1010  REM  UPDATE MULTIPLIER
1020 LF = LG: IF P2 = 0 THEN LF = 1
1021  IF TS%(LF) = 0 THEN 1040
1022  FOR J = 1 TO TS%(LF): IF SK$ = S$(J,LF) THEN 1060
1030  NEXT J
1040 TS%(LF) = TS%(LF) + 1:J = TS%(LF):SC%(J,LF) = 0:S$(J,LF) = SK$
1050  INVERSE : PRINT " * NEW MULTIPLIER * ": NORMAL
1060 SC%(J,LF) = SC%(J,LF) + 1: RETURN
1070  REM  PRINTER INIT, IF USED
1080  RETURN
1090  PRINT LP$: PRINT : PRINT : PRINT  SPC( 4)"CONTEST: ";Y$
1100  PRINT  SPC( 4)"DATE:     ";Z$
1110  PRINT : PRINT SCN$: RETURN
1120  REM  SET DATE
1130  IF RC$ = "Y" THEN 1162
1132  GOSUB 2400
1140  IF G$ = "" THEN  PRINT : PRINT "WHAT IS THE DATE (MM/DD/YY) ";: GOSUB 2370:
GOTO 1140
1150  IF  LEN (G$) > 8 THEN  PRINT "DATE TOO LONG";BL$:G$ = "": GOTO 1140
1160 DTE$ = G$: RETURN
1162  PRINT D$;"IN#1": INPUT "DATE/TIME = ";G$: PRINT D$;"IN#0":DTE$ =  MID$ (G$,3
,8):UTC$ =  MID$ (G$,12,2) +  MID$ (G$,15,2)
1164  RETURN
```

```
1170  REM  SET MODE
1180  GOSUB 2400
1190  IF G$ = "" THEN  PRINT : INPUT "WHAT IS THE MODE? ";G$
1200  IF  LEN (G$) > 3 THEN  PRINT "INVALID MODE. 3 CHARACTERS MAX.";BL$:G$ = "":
GOTO 1190
1210 MD$ = G$: RETURN
1220  REM  X DELETES LAST ENTRY
1230  IF LH% = 0 THEN 660
1240  IF LH% < SS THEN  IF P%(LH%) = 0 THEN C1%(LH%) =  - 32000: GOTO 1260
1250 P%(LH%) = 0:L% = L% - 1
1260 T = T - 1:TC%(LG) = TC%(LG) - 1:LH% = 0
1270  GOSUB 1310
1280  IF HC$ = "Y" THEN  PRINT LP$: PRINT "-- PREVIOUS CALL DELETED --": PRINT SCN
$
1290  PRINT " DELETED": RETURN
1300  REM  DELETE SECTION
1310 LF = LG: IF P2 = 0 THEN LF = 1
1312  FOR J = 1 TO TS%(LF): IF SK$ = S$(J,LF) THEN 1330
1320  NEXT J
1330 SC%(J,LF) = SC%(J,LF) - 1: IF SC%(J,LF) > 0 THEN  RETURN
1340  IF J = TS%(LF) THEN 1360
1350 SC%(J,LF) = SC%(TS%(LF),LF):S$(J,LF) = S$(TS%(LF),LF)
1360 TS%(LF) = TS%(LF) - 1: RETURN
1370  REM  MENU
1380  HOME : PRINT : INVERSE
1390  PRINT "    GENERAL CONTEST LOGGER MENU          "
1400  PRINT "                                         "
1410  PRINT " '?' DISPLAYS THIS MENU                  "
1420  PRINT " 'QSY <BAND> <MODE>' CHANGES BANDS       "
1430  PRINT " 'DATE <MM/DD/YY>' CHANGES DATE          "
1440  PRINT " 'MODE <MODE>' CHANGES MODES             "
1450  PRINT " 'QRT' SAVES ALL CALLS AND EXITS         "
1455  PRINT " 'QRX' SAVES ALL CALLS AND STOPS TIMER   "
1460  PRINT " 'SAVE' SAVES ALL CALLS AND RESUMES      "
1470  PRINT "        (NOTE: ALLOW 2-3 MIN FOR SAVE)   "
1475  PRINT " 'RATE' SHOWS TOTAL TIME USED AND RATE   "
1480  PRINT " 'STAT' GIVES STATUS ON CURRENT BAND     "
1490  PRINT " 'SCORE' GIVES RUNNING TOTAL SCORE       "
1500  PRINT " 'EDIT <QSO #>' EDITS ANY LOGGED ENTRY   "
1510  PRINT " 'X' DELETES THE LAST ENTRY              "
1520  PRINT " '-' ABORTS CURRENT ENTRY                "
1530  PRINT " ENTRY FORMAT: 'CALL EXCHANGE <UTC>'     "
1540  PRINT " ENTER CALL ONLY FOR DUPE CHECK          "
1550  PRINT " ENTER CALL AND '?' TO LIST BANDS        "
1552  PRINT "       ON WHICH A STATION IS NEEDED      "
1560  PRINT " <> DENOTES OPTIONAL ENTRIES             "
1570  NORMAL : RETURN
1580  REM  QSY
1590  GOSUB 2400
1600  IF NL = 1 THEN  RETURN
1610  IF G$ = "" THEN  PRINT : INPUT "WHAT BAND (LOG SHEET) DO YOU WANT? ";G$
1620  FOR J = 1 TO NL: IF G$ = LN$(J) THEN 1660
1630  NEXT J
1640  PRINT "NO SUCH LOG. VALID LOGS ARE:";BL$
1650  FOR J = 1 TO NL: PRINT  TAB( 5)LN$(J): NEXT J:G$ = "": GOTO 1610
1660 LG = J: IF HC$ = "Y" THEN  GOSUB 2490
1670  GOSUB 2400: IF G$ <  > "" THEN  GOSUB 1200
1680  RETURN
1690  REM  STATUS
1700  HOME : PRINT : PRINT "DATE= ";DTE$;
1710  PRINT "  BAND=";LN$(LG);
1720  PRINT "  MODE= ";MD$
1725 LF = LG: IF P2 = 0 THEN LF = 1
1730  PRINT " # OF M'PLRS= ";TS%(LF);
1740  PRINT "  CONTACTS= ";TC%(LG)
1750  PRINT : RETURN
1760  REM  LIST SECTIONS
1770  HOME : PRINT :LF = LG: IF P2 = 0 THEN LF = 1
1780  PRINT "------ MULTIPLIERS WORKED";: IF P2 = 1 THEN  PRINT " ON ";LN$(LF);
```

```
1782  PRINT " ------"
1790  IF TS%(LF) = 0 THEN  RETURN
1800 TB = 1
1810  FOR J = 1 TO TS%(LF)
1820  PRINT  TAB( TB)S$(J,LF);
1830 TB = TB + 8: IF TB > 39 THEN TB = 1: PRINT
1840  NEXT J: PRINT : RETURN
1850  REM  QRT
1860  PRINT OP$;F1$;DV$
1870  PRINT DL$;F1$
1880  PRINT OP$;F1$
1890  PRINT WR$;F1$
1900  PRINT HC$: PRINT T: PRINT Y$: PRINT Z$: PRINT NL: PRINT L%: PRINT M1
1910  FOR J = 1 TO NL: PRINT LN$(J): NEXT J
1912  PRINT ZA: PRINT ET: FOR J = 1 TO ZA: PRINT U1$(J): PRINT U2$(J): PRINT U3(J)
: NEXT J
1950  FOR J = 1 TO NL: PRINT QP%(J): PRINT TC%(J): PRINT TS%(J): NEXT J
1960  FOR J = 1 TO NL: FOR K = 1 TO TS%(J)
1970  PRINT SC%(K,J): PRINT S$(K,J)
1980  NEXT K: NEXT J
1982  PRINT P1$: PRINT P2
1985  RETURN
1990  REM  WRITE SECT SUMMARY
2000  PRINT BL$: REM  NEW PAGE
2010  PRINT Q$; SPC( 27)"* MULTIPLIER SUMMARY *";Q$
2015 LF = NL: IF P2 = 0 THEN LF = 1
2020  FOR J = 1 TO LF
2030  PRINT : PRINT : PRINT
2040  PRINT "----------------------  MULTIPLIERS";: IF P2 = 1 THEN  PRINT " WORKED
 ON ";LN$(J);
2042  PRINT "  ----------------------": PRINT
2050  PRINT Q$;: FOR K = 1 TO 5: PRINT "  M'PLIER/QSOS ";: NEXT K: PRINT Q$
2060 TB = 9
2070  FOR K = 1 TO TS%(J)
2075  IF TB = 9 THEN  PRINT Q$;
2080  PRINT  SPC( 9 -  LEN (S$(K,J)))S$(K,J);"/";SC%(K,J); SPC( 5 -  LEN ( STR$ (S
C%(K,J))));
2090 TB = TB + 15: IF TB > 74 THEN TB = 9: PRINT Q$
2100  NEXT K: IF TB > 9 THEN  PRINT Q$
2105  PRINT : NEXT J
2110  REM  WRITE SCORE
2120  PRINT BL$: REM  NEW PAGE
2130  PRINT Q$; SPC( 13)"SCORE SUMMARY";Q$: PRINT : PRINT Q$;T1$;Q$
2150 X2 = 0:X3 = 0:X4 = 0
2160  FOR J = 1 TO NL
2170 X4 = X4 + TC%(J):X1 = TC%(J) * QP%(J):X2 = X2 + X1:X3 = X3 + TS%(J)
2180  PRINT LN$(J); SPC( 10 -  LEN (LN$(J)))TC%(J); SPC( 9 -  LEN ( STR$ (TC%(J)))
)QP%(J); SPC( 7 -  LEN ( STR$ (QP%(J))))X1;
2190  IF P2 = 1 THEN  PRINT  SPC( 10 -  LEN ( STR$ (X1)))TS%(J);
2192  PRINT
2200  NEXT J
2210  PRINT Q$;T2$;Q$
2212  IF P2 = 0 THEN X3 = TS%(1)
2215  PRINT "TOTAL      ";X4; SPC( 16 -  LEN ( STR$ (X4)))X2; SPC( 10 -  LEN ( STR$
 (X2)))X3: PRINT
2230  PRINT Q$; SPC( 24)"***********";Q$
2240  PRINT Q$; SPC( 24)"*          *";Q$
2250  PRINT "SCORE = ";X2;" X ";X3; SPC( 13 -  LEN ( STR$ (X2)) -  LEN ( STR$ (X3)
))"* ";X2 * X3; SPC( 8 -  LEN ( STR$ (X2 * X3)))"*"
2260  PRINT Q$; SPC( 24)"*          *";Q$
2270  PRINT Q$; SPC( 24)"***********";Q$
2275  PRINT "!": REM  EOF MARK
2280  PRINT CL$
2281  PRINT OP$;F2$;DV$
2282  PRINT DL$;F2$
2283  PRINT OP$;F2$;R2$;DV$
2284  FOR J = 1 TO L%
2285  PRINT WR$;F2$;",R";J
2286  PRINT C1%(J): PRINT C2%(J): PRINT P%(J)
2287  NEXT J
```

```
2288  PRINT CL$: RETURN
2290  IF HC$ = "Y" THEN  PRINT LP$: GOSUB 3675: PRINT SCN$
2300  END
2310  HOME : PRINT : PRINT : INVERSE : REM  PRINT TITLE BANNER
2320  PRINT "                                        "
2330  PRINT "    GENERAL CONTEST LOGGING PROGRAM     "
2340  PRINT "    (C) 1983 BY HAYDEN BOOK CO., INC.   "
2345  PRINT "                                        "
2350  NORMAL : PRINT : RETURN
2360  REM  INPUT SCAN ROUTINE
2370  INPUT GS$:LL =  LEN (GS$): IF LL = 0 THEN 2430
2380  FOR J = 1 TO LL: IF  MID$ (GS$,J,1) = "-" THEN X1% = 1: GOTO 650
2381  NEXT J
2390 LP = 1
2400  IF LP > LL THEN 2430: REM  2ND ENTRY
2410  FOR LT = LP TO LL: IF  MID$ (GS$,LT,1) <  > " " THEN 2440
2420  NEXT LT
2430 G$ = "": RETURN
2440 LP = LT
2450  FOR LT = LP TO LL: IF  MID$ (GS$,LT,1) = " " THEN 2480
2460  NEXT LT
2470 LT = LT + 1
2480 G$ =  MID$ (GS$,LP,LT - LP):LP = LT: RETURN
2490  PRINT LP$: PRINT
2500  PRINT  TAB( 5)"QSY TO: ";LN$(LG)
2510  PRINT SCN$: RETURN
2530  REM  RESUME
2540  PRINT OP$;F1$;DV$
2550  PRINT RD$;F1$
2570  INPUT HC$: INPUT T: INPUT Y$: INPUT Z$: INPUT NL: INPUT L%: INPUT M1
2580  DIM LN$(NL),QP%(NL),TC%(NL),TS%(NL),SC%(100,NL)
2590  DIM C1%(M1),C2%(M1),P%(M1),S$(100,NL)
2600  FOR J = 1 TO NL: INPUT LN$(J): NEXT J
2602  INPUT ZA: INPUT ET: FOR J = 1 TO ZA: INPUT U1$(J): INPUT U2$(J): INPUT U3(J)
: NEXT J
2610  FOR J = 1 TO NL: INPUT QP%(J): INPUT TC%(J): INPUT TS%(J): NEXT J
2620  FOR J = 1 TO NL: FOR K = 1 TO TS%(J)
2630  INPUT SC%(K,J): INPUT S$(K,J)
2640  NEXT K: NEXT J
2642  INPUT P1$: INPUT P2
2645  PRINT CL$
2650  PRINT OP$;F2$;R2$;DV$
2660  FOR J = 1 TO L%
2670  PRINT RD$;F2$;",R";J
2680  INPUT C1%(J): INPUT C2%(J): INPUT P%(J)
2682  NEXT J
2690  PRINT CL$: RETURN
2700  REM  EDIT A RECORD
2710  GOSUB 2400: IF G$ = "" THEN  INPUT "WHAT QSO# DO YOU WANT TO EDIT? ";G$
2720 X2 =  INT ( VAL (G$))
2730  IF X2 > 0 THEN  IF X2 < T THEN 2750
2740  GOTO 670
2750  HOME :JB = 8: IF P1$ = "Y" THEN JB = 7
2760  PRINT OP$;F3$;R3$;DV$
2761  PRINT RD$;F3$;",R";X2
2762  INPUT G$: PRINT CL$
2763 Y$(2) =  LEFT$ (G$,10):Y$(3) =  MID$ (G$,11,6):Y$(4) =  MID$ (G$,17,12)
2764 Y$(5) =  MID$ (G$,29,7):Y$(6) =  MID$ (G$,36,7):Y$(7) =  RIGHT$ (G$,15)
2765  FOR J = 2 TO 7:GS$ = Y$(J):LL =  LEN (GS$): GOSUB 2390:Y$(J + 10) = G$:Y$(J)
 = G$: NEXT J
2766 Y$(8) = Y$(7): IF P1$ <  > "Y" THEN  GOSUB 2400:Y$(8) = G$
2767 Y$(18) = Y$(8)
2770  GOSUB 3080
2772  FOR J = 2 TO JB: VTAB (J + 1): HTAB (14): PRINT Y$(J): NEXT J
2780  VTAB (11): INPUT "CHANGE THIS RECORD (Y/N)? ";A$
2790  IF A$ <  > "Y" THEN  HOME : GOTO 450
2800 LH% = 0
2810  FOR J = 1 TO NL: IF Y$(5) = LN$(J) THEN 2830
2820  NEXT J
2830 LX = LG:LY = J:LK = LY
```

```
2840  VTAB (11): INVERSE
2842  PRINT "            INSTRUCTIONS                "
2844  PRINT "                                        "
2850  PRINT "HIT <CR> TO LEAVE AN ITEM UNCHANGED.    "
2860  PRINT "TO CHANGE, ENTER THE NEW VALUE <CR>.    "
2870  PRINT "TO DELETE THIS ENTRY, CHANGE THE CALL   "
2871  PRINT "      TO '*'.                           "
2872  NORMAL
2880  FOR J = 2 TO JB
2890  VTAB (J + 1): HTAB (14): INPUT "";G$: IF G$ < > "" THEN Y$(J) = G$
2895  VTAB (J + 1): HTAB (14): PRINT Y$(J)
2900  NEXT J
2910  IF P1$ = "Y" THEN Y$(8) = Y$(7)
2940  VTAB (18): INPUT "OK (Y/N)? ";A$:X1% = 4
2942  PRINT
2950  IF A$ < > "Y" THEN  HOME : GOSUB 650: GOTO 450
2960  IF Y$(5) < > Y$(15) THEN  GOSUB 3240: GOTO 3000: REM  BAND CHANGE
2970 LG = LK
2980  IF Y$(4) < > Y$(14) THEN  GOSUB 3160: GOSUB 3190: REM  CALL CHANGE
2990  IF Y$(8) < > Y$(18) THEN  GOSUB 3320: GOSUB 3350: REM  EXCHANGE CHANGE
3000 LG = LX
3010  ON X1% GOTO 590,3370,3370,3020,3020,3380
3020  PRINT OP$;F3$;R3$;DV$
3030  PRINT WR$;F3$;",R";X2
3035  IF P1$ = "Y" THEN Y$(7) = ""
3040  PRINT Y$(2);
3041  PRINT  SPC( 10 -  LEN (Y$(2)))Y$(3);
3042  PRINT  SPC( 6 -  LEN (Y$(3)))Y$(4);
3043  PRINT  SPC( 12 -  LEN (Y$(4)))Y$(5);
3044  PRINT  SPC( 7 -  LEN (Y$(5)))Y$(6);
3045  PRINT  SPC( 7 -  LEN (Y$(6)))Y$(7);" ";Y$(8);
3046  PRINT  SPC( 14 -  LEN (Y$(7)) -  LEN (Y$(8)))""
3048  PRINT CL$
3050  IF HC$ < > "Y" THEN 3070
3060  PRINT LP$: PRINT "  ** EDIT ON REC #";X2;":";
3062  FOR J = 2 TO 8: PRINT " ";Y$(J);: NEXT J: PRINT
3065  PRINT SCN$
3070  GOTO 450
3080  PRINT  TAB( 5)"EDITING RCD #";X2: PRINT : PRINT  TAB( 5)"DATE:"
3090  PRINT  TAB( 5)"TIME:"
3100  PRINT  TAB( 5)"CALL:"
3110  PRINT  TAB( 5)"BAND:"
3120  PRINT  TAB( 5)"MODE:"
3130  IF P1$ < > "Y" THEN  PRINT  TAB( 5)"EXCHANGE:"
3135  PRINT  TAB( 5)"M'PLIER:"
3140  RETURN
3150  REM  CALL CHANGED
3160  IF Y$(4) = "*" THEN Y$(8) = "*": RETURN
3170 CS$ = Y$(4): GOSUB 3400: IF X1% < 4 THEN  RETURN : REM  CHECK IF NEW CALL IS
DUPE
3180  GOSUB 990:TC%(LG) = TC%(LG) + 1: RETURN
3190  IF X1% < 4 THEN  RETURN
3195  IF Y$(14) = "*" THEN  RETURN
3200 CS$ = Y$(14)
3210  GOSUB 3400:C1%(H%) =  - 32001:C2%(H%) =  - 32000: REM  DELETE OLD CALL
3220 TC%(LG) = TC%(LG) - 1:X1% = 4: RETURN
3230  REM  CHANGE BAND
3240  FOR LK = 1 TO NL: IF Y$(5) = LN$(LK) THEN 3270
3250  NEXT LK
3260 X1% = 6: RETURN
3270 LG = LK: GOSUB 3160: IF X1% < 4 THEN  RETURN
3280 LG = LY: GOSUB 3190
3290  GOSUB 3320:LG = LK: GOSUB 3350: RETURN
3300 LG = LY: GOSUB 3350: RETURN
3310  REM  EXCH CHANGE
3320  IF Y$(18) = "*" THEN  RETURN
3330 SK$ = Y$(18)
3340  GOTO 1310: REM  DELETE OLD SECTION
3350  IF Y$(8) = "*" THEN  RETURN
3360 SK$ = Y$(8): GOSUB 1020: RETURN : REM  ADD NEW SECTION
3370  PRINT BL$;"INVALID CALL SIGN": GOTO 450
```

```
3380  PRINT BL$;"INVALID BAND": GOTO 450
3390  REM  DUPE CHECKER:X1%=1=DUPE: 2=QUERY: 3=INVALID CALL: 4,5=OK
3400 X1% = 4:E(5) = 0:E(6) = 0:K = 1: IF  LEN (CS$) > 10 THEN X1% = 3: RETURN
3410  IF  RIGHT$ (CS$,1) = "?" THEN CS$ =  LEFT$ (CS$, LEN (CS$) - 1):X1% = 2:CT$
= CS$
3420  IF  LEN (CS$) < 4 THEN X1% = 3: RETURN
3430  IF  MID$ (CS$,3,1) < "A" THEN 3460
3440  IF  MID$ (CS$,2,1) =  > "A" THEN X1% = 3: RETURN
3450 CS$ = " " + CS$:K = 2:E(1) = 0
3460  FOR J = K TO  LEN (CS$)
3470 W$ =  MID$ (CS$,J,1): IF W$ = "/" THEN 3520
3480 E(J) =  ASC (W$) - AA: IF E(J) > 26 THEN X1% = 3: RETURN
3490  IF W$ < "0" THEN X1% = 3: RETURN
3500  IF E(J) < 1 THEN E(J) = E(J) + A0: IF E(J) > 36 THEN X1% = 3: RETURN
3510  NEXT J
3520  IF J > 7 THEN X1% = 3: RETURN
3530  IF J - K < 4 THEN X1% = 3: RETURN
3540  FOR J = 4 TO 6: IF E(J) > 26 THEN X1% = 3: RETURN
3550  NEXT J
3560 E1% =  - 32000 + E(1) * 1369 + E(2) * 37 + E(3)
3570 E2% =  - 32000 + E(4) * 1369 + E(5) * 37 + E(6)
3580  IF X1% = 2 THEN  RETURN
3590 H% = E(4) + LG * 26 - 26
3600  REM  SEARCH DUPE STRING
3610  IF C1%(H%) =  - 32000 THEN X1% = 4: RETURN
3620  IF C2%(H%) <  > E2% THEN 3640
3630  IF C1%(H%) = E1% THEN X1% = 1: RETURN
3640  IF P%(H%) = 0 THEN X1% = 5: RETURN
3650 H% = P%(H%): GOTO 3620
3660  REM  SCORE
3675  PRINT : PRINT  SPC( 12)"SCORE SUMMARY": PRINT
3680  PRINT T1$
3690 X2 = 0:X3 = 0:X4 = 0
3700  FOR J = 1 TO NL
3710 X4 = X4 + TC%(J):X1 = TC%(J) * QP%(J):X2 = X2 + X1:X3 = X3 + TS%(J)
3720  PRINT LN$(J);
3721  PRINT  SPC( 10 -  LEN (LN$(J)))TC%(J);
3722  PRINT  SPC( 9 -  LEN ( STR$ (TC%(J))))QP%(J);
3723  PRINT  SPC( 7 -  LEN ( STR$ (QP%(J))))X1;
3724  IF P2 = 1 THEN  PRINT  SPC( 9 -  LEN ( STR$ (X1)))TS%(J);
3726  PRINT
3730  NEXT J
3740  PRINT T2$
3745  IF P2 = 0 THEN X3 = TS%(1)
3750  PRINT "TOTAL     ";X4;
3751  PRINT  SPC( 16 -  LEN ( STR$ (X4)))X2;
3752  PRINT  SPC( 9 -  LEN ( STR$ (X2)))X3
3760  PRINT : PRINT  SPC( 9)"SCORE = ";X3 * X2
3770  PRINT : RETURN
3900  PRINT DTE$;
3901  PRINT  SPC( 10 -  LEN (DTE$))UTC$;
3902  PRINT  SPC( 6 -  LEN (UTC$))CT$;
3903  PRINT  SPC( 12 -  LEN (CT$))LN$(LG);
3904  PRINT  SPC( 7 -  LEN (LN$(LG)))MD$;
3905  PRINT  SPC( 7 -  LEN (MD$))SL$;" ";SK$;
3906  PRINT  SPC( 14 -  LEN (SK$) -  LEN (SL$))""
3910  RETURN
3920  IF RC$ <  > "Y" THEN 3925
3922  GOSUB 1162:U2$(ZA) = UTC$: GOTO 3930
3925  PRINT : INPUT "ENTER THE TIME IN UTC";U2$(ZA)
3930 UX =  VAL (U1$(ZA)):UY =  VAL (U2$(ZA))
3932 U7 =  INT (UX / 100):U4 = ((UX / 100) - U7) / .6:U7 = U7 + U4
3940 U5 =  INT (UY / 100):U6 = ((UY / 100) - U5) / .6:U5 = U5 + U6
3950  IF U5 < U7 THEN U5 = U5 + 24
3960 E7 = ET:ET = ET + U5 - U7:U3(ZA) = T: IF ET = 0 THEN 3990
3970  PRINT : PRINT "TIME USED SO FAR: "; INT (ET);" HRS, "; INT ((ET -  INT (ET))
 * 60 + .5);" MINS"
3980  PRINT : PRINT "OVERALL RATE: "; INT ((T - 1) / ET + .5);" QSOS PER HOUR"
3990  RETURN
```

GENERAL CONTEST LOGGER (Commodore 64)

```
10 REM GENERAL PURPOSE CONTEST LOGGER (C64)
15 REM DESIGNED FOR DX/VHF CONTESTS, QSO PARTIES, ETC.
20 REM BY KC6A AND N6NB
30 REM COPYRIGHT 1983 BY HAYDEN BOOK CO.,INC.
35 OPEN 130,2,2,CHR$(136)
36 OPEN15,8,15
40 BL$=CHR$(7):REM BELL (DON'T DELETE)
50 C7$=CHR$(147):REM CLEAR SCREEN COMMAND
52 HI$=CHR$(18):LO$=CHR$(146)
54 REM PUT YOUR HIGHLIGHTING AND NORMAL VIDEO COMMANDS IN LINE 52
60 FOR J=1 TO 5:PRINT:NEXT J
70 FA$="0:DUPELOGA":FB$="0:DUPELOGB":FC$="0:CONTSTLG"
80 ET=0:ZA=0:AA=ASC("A")-1:A0=ASC("A")-ASC("0")+26
90 T1$="BAND      QSO'S X Q-PTS = POINTS  M'PLRS"
100 T2$="----      -----          ------    ---"
105 DIM Y$(20),U1$(15),U2$(15),U3(15)
110 GOSUB 2310
115 PRINT:PRINT "DO YOU WANT TO USE THE 'JIFFY CLOCK' TO"
116 INPUT "LOG THE TIME (Y/N) ";RC$
117 IF RC$<>"Y" THEN 120
118 INPUT "ENTER THE TIME IN UTC (FORMAT: 0000)";T9$
119 TI$=T9$+"00"
120 PRINT:INPUT "NEW CONTEST (N) OR RESUME (R) ";A$
130 IF A$="N" THEN 190
140 IF A$<>"R" THEN PRINT BL$:GOTO 110
150 GOSUB 2540:REM READ IN FILE
155 PRINT:PRINT "RESUMING ";Y$;" OF ";Z$
160 IF HC$="Y" THEN GOSUB 1070
170 SS=26*NL+1:GOTO 380
180 REM NEW CONTEST
190 PRINT#15,"S"+FA$:PRINT#15,"S"+FB$:PRINT#15,"S"+FC$
195 PRINT:PRINT "WANT HARD COPY WHILE LOGGING (Y/N)"
196 HC$="":INPUT HC$
200 IF HC$="Y" THEN INPUT "TURN ON PRINTER. <CR> WHEN READY ";AA$:GOSUB 1070
210 PRINT:INPUT "WHAT IS THE CONTEST NAME ";Y$
215 PRINT:PRINT "NOTE: FOR CONTESTS IN WHICH YOU CANNOT"
216 PRINT "WORK STATIONS AGAIN ON ADDITIONAL BANDS"
217 PRINT "SPECIFY ONE LOG HERE AND USE THE "
218 PRINT "'MODE' SPECIFIER TO RECORD BAND CHANGES"
220 PRINT:PRINT "HOW MANY LOGS DO YOU NEED"
221 INPUT "  (USUALLY ONE PER BAND) ";NL:PRINT
230 IF NL<1 THEN PRINT BL$:GOTO 220
240 DIM LN$(NL),QP%(NL),TC%(NL),TS%(NL),SC%(100,NL)
250 M1=INT((FRE(0)-2000-800*NL)/6)
260 IF NL*26>=M1 THEN PRINT "TOO MANY";BL$:ERASE LN$,QP%,TC%,TS%,SC%:GOTO 220
270 DIM C1%(M1),C2%(M1),P%(M1),S$(100,NL)
275 KK$=""
280 FOR J=1 TO NL
290 PRINT:PRINT "WHAT BAND IS LOG SHEET #";J
300 PRINT "  (E.G. 80,20,144,1296 OR EVEN 10 GHZ"
302 INPUT LN$(J)
310 IF LEN(LN$(J))>8 THEN PRINT BL$;"NAME TOO LONG":GOTO 290
314 IF J=1 THEN 320
315 FOR K=1 TO J-1:IF LN$(J)=LN$(K) THEN PRINT "DUPLICATE BAND";BL$:GOTO 290
316 NEXT K
320 PRINT:INPUT "HOW MANY POINTS PER QSO ";QP%(J)
330 IF QP%(J)<1 THEN PRINT BL$:GOTO 320
331 KK$=KK$+LN$(J)+"  "
332 PRINT:PRINT "BANDS SPECIFIED: ";KK$:PRINT
340 TC%(J)=0:TS%(J)=0:NEXT J
350 T=1:L%=26*NL:SS=L%+1
352 P2=0:IF NL=1 THEN GOTO 355
353 PRINT:PRINT"SEPARATE MULTIPLIERS ON":INPUT" EACH BAND (Y/N) ";G$
354 IF G$="Y" THEN P2=1
355 PRINT:PRINT "YOU HAVE TWO OPTIONS FOR THE EXCHANGE:"
356 PRINT "THE ENTIRE EXCHANGE CAN BE A MULTIPLIER"
357 PRINT "OR OTHER INFO CAN BE SET OFF FROM THE"
```

```
358 PRINT "MULTIPLIER WITH A SPACE (E.G. '59 ENY')"
359 PRINT:INPUT "IS THE WHOLE EXCHANGE THE MULT (Y/N)";P1$
360 IF P1$<>"N" THEN 365
361 PRINT:PRINT "BE SURE TO PUT A SPACE BETWEEN"
362 PRINT "THE REPORT AND THE MULTIPLIER"
365 PRINT:PRINT " MEMORY IS AVAILABLE FOR ABOUT"
366 PRINT M1;" CONTACTS, BUT DISK SPACE IS"
367 PRINT " LIMITED TO ABOUT 1200 QSOS ON THIS"
368 PRINT " COMPUTER AND DISK OPERATING SYSTEM."
369 PRINT:PRINT TAB(15)"GOOD LUCK!"
370 FOR J=1 TO L%:C1%(J)=-32000:P%(J)=0:NEXT J
380 GOSUB 3780
390 LG=1:LH%=0:LP=1:LL=0:SL$=""
400 GOSUB 1132:IF A$="N" THEN Z$=DTE$:REM DATE
410 IF HC$="Y" THEN GOSUB 1090
420 GOSUB 1590:REM BAND
430 GOSUB 1180:REM MODE
440 GOSUB 1380:REM MENU
442 ZB$="":INPUT "<CR> TO LOG STARTING TIME";ZB$
443 ZA=ZA+1:IF RC$<>"Y" THEN 446
444 GOSUB 1162:U1$(ZA)=UTC$:GOTO 450
446 INPUT "ENTER UTC TIME (0000 FORMAT)";U1$(ZA)
450 IF T/50<>INT(T/50) THEN 454
451 PRINT:INPUT "TO RESET CLOCK, ENTER UTC (0000 FORMAT)";T9$
452 TI$=T9$+"00"
454 PRINT:PRINT "LOGGER READY FOR QSO #";T;"-";LN$(LG);"-";MD$
460 PRINT "COMMAND OR CALL";
470 X1%=0:GOSUB 2370:IF X1%=1 THEN 450
480 IF G$="?" THEN GOSUB 1380:GOTO 450
490 IF G$="QSY" THEN GOSUB 1590:GOSUB 1700:GOTO 450
500 IF G$="DATE" THEN GOSUB 1132:GOTO 450
510 IF G$="MODE" THEN GOSUB 1180:GOTO 450
520 IF G$="QRT" THEN GOSUB 3900:GOSUB 3670:GOSUB 1860:GOTO 2000
525 IF G$="QRX" THEN GOSUB 3900:GOSUB 1860:CLOSE4:GOSUB 3800:GOTO 442
530 IF G$="SAVE" THEN GOSUB 1860:CLOSE4:GOSUB 3800:GOTO 450
535 IF G$="RATE" THEN GOSUB 3900:ET=E7:GOTO 450
540 IF G$="STAT" THEN GOSUB 1700:GOSUB 1780:GOTO 450
550 IF G$="SCORE" THEN GOSUB 3670:GOTO 450
560 IF G$="EDIT" THEN GOTO 2710
570 IF G$="X" THEN GOSUB 1230:GOTO 450
580 GOTO 690
590 PRINT BL$
600 PRINT HI$;"****************************************"
610 PRINT HI$;"**  IT'S A DUPE  **  BOO  **  HISS  **"
620 PRINT HI$;"****************************************";LO$
630 GOTO 450
640 PRINT "INVALID ENTRY. PLEASE TRY AGAIN.";BL$:GOTO 450
650 PRINT "ABORTED":RETURN
660 PRINT "CAN'T DELETE";BL$:RETURN
670 PRINT "INVALID QSO #";BL$:GOTO 450
680 REM PROCESS A CALL SIGN
690 CS$=G$:CT$=G$
700 GOSUB 3400:ON X1% GOTO 590,820,640,710,710:REM DECODE CALL/JUMP ON RESULT
710 GOSUB 2400:IF LEN(G$)<>0 THEN 730
720 PRINT " -OK- ";
730 IF P1$="Y" THEN 740
732 SL$=G$:IF X1%=1 THEN 450
734 IF SL$="" THEN PRINT "REPORT (OR FULL EXCH)? ";:GOSUB 2370::GOTO 732
735 GOSUB 2400
740 SK$=G$:IF X1%=1 THEN 450
742 IF SK$="" THEN PRINT "MULTIPLIER? ";:GOSUB 2370:GOTO 740
750 IF LEN(SK$)+LEN(SL$)<=14 THEN 752
751 PRINT BL$;"EXCHANGE EXCEEDS 14 CHARACTERS":PRINT:G$="":GOTO 730
752 IF LEN (SK$)<1 THEN PRINT BL$:PRINT:G$="":GOTO 740
760 IF RC$="Y"THEN GOSUB 1162:GOTO 770
762 GOSUB 2400:UTC$=G$
766 IF UTC$="" THEN PRINT "UTC? ";:GOSUB 2370:UTC$=G$:IF X1%=1 THEN 450
770 RN=T:GOSUB 900:REM WRITE LOG TO DISK
780 IF HC$="Y" THEN GOSUB 960:REM WRITE TO PRINTER
790 LH%=H%:GOSUB 990:REM ENTER IN DUPE SHEET
```

```
800 GOSUB 1020:REM ENTER SECTION
810 TC%(LG)=TC%(LG)+1:T=T+1:GOTO 450
820 PRINT:PRINT TAB(2)CT$;" NEEDED ON:"
830 FOR J=1 TO NL
840 H%=E(4)+J*26-26:GOSUB 3610:REM DUPE SEARCH
850 IF X1%=1 THEN 880
860 IF J=LG THEN PRINT TAB(10)"* ";:GOTO 870
862 PRINT TAB(11)" ";
870 PRINT LN$(J)
880 NEXT J:PRINT:GOTO 450
890 REM WRITE LOG TO DISK
900 PRINT "...LOGGING"
910 GOSUB 7000
912 RP=1:GOSUB 7100:PRINT#3,DTE$
914 RP=10:GOSUB 7100:PRINT#3,UTC$
916 RP=15:GOSUB 7100:PRINT#3,CT$
918 RP=26:GOSUB 7100:PRINT#3,LN$(LG)
920 RP=32:GOSUB 7100:PRINT#3,MD$
922 RP=38:GOSUB 7100:PRINT#3,SL$+" "+SK$
930 GOSUB 6000:RETURN
950 REM WRITE LOG TO PRINTER
960 PRINT#130,T;SPC(2)DTE$;SPC(3)UTC$;SPC(3)LN$(LG);SPC(8-LEN(LN$(LG)))MD$;
962 PRINT#130," ";SPC(6-LEN(MD$))CT$;SPC(12-LEN(CT$))SL$;" ";SK$
970 RETURN
980 REM ENTER IN DUPE LIST
990 IF X1%=5 THEN L%=L%+1:P%(H%)=L%:H%=L%
1000 C1%(H%)=E1%:C2%(H%)=E2%:P%(H%)=0:RETURN
1010 REM UPDATE SECTION
1020 LF=LG:IF P2=0 THEN LF=1
1022 FOR J=1 TO TS%(LF):IF SK$=S$(J,LF) THEN 1060
1030 NEXT J
1040 TS%(LF)=TS%(LF)+1:J=TS%(LF):SC%(J,LF)=0:S$(J,LF)=SK$
1050 PRINT " * NEW MULTIPLIER *"
1060 SC%(J,LF)=SC%(J,LF)+1:RETURN
1070 REM PRINTER INIT, IF USED
1080 RETURN
1090 PRINT#130,"":PRINT#130,"":PRINT#130,"      CONTEST: ";Y$
1100 PRINT#130,"      DATE:    ";Z$
1110 PRINT#130,"":RETURN
1120 REM SET DATE
1130 IF RC$="Y" THEN 1162
1132 GOSUB 2400
1140 IF G$="" THEN PRINT:INPUT "WHAT IS THE DATE (MM/DD/YY) ";G$
1150 IF LEN(G$)>8 THEN PRINT "DATE TOO LONG";BL$:G$="":GOTO 1140
1160 DTE$=G$:RETURN
1162 UTC$=LEFT$(TI$,4)
1164 RETURN
1170 REM SET MODE
1180 GOSUB 2400
1190 IF G$="" THEN PRINT:INPUT "WHAT IS THE MODE ";G$
1200 IF LEN(G$)<=3 THEN 1210
1201 PRINT "INVALID MODE. 3 CHARACTERS MAX.";BL$:G$="":GOTO 1190
1210 MD$=G$:RETURN
1220 REM X DELETES LAST ENTRY
1230 IF LH%=0 THEN 660
1240 IF LH%<SS THEN IF P%(LH%)=0 THEN C1%(LH%)=-32000:GOTO 1260
1250 P%(LH%)=0:L%=L%-1
1260 T=T-1:TC%(LG)=TC%(LG)-1:LH%=0
1270 GOSUB 1310
1280 IF HC$="Y" THEN PRINT#130,"-- PREVIOUS CALL DELETED --"
1290 PRINT " DELETED":RETURN
1300 REM DELETE SECTION
1310 LF=LG:IF P2=0 THEN LF=1
1312 FOR J=1 TO TS%(LF):IF SK$=S$(J,LF) THEN 1330
1320 NEXT J
1330 SC%(J,LF)=SC%(J,LF)-1:IF SC%(J,LF)>0 THEN RETURN
1340 IF J=TS%(LF) THEN 1360
1350 SC%(J,LF)=SC%(TS%(LF),LF):S$(J,LF)=S$(TS%(LF),LF)
1360 TS%(LF)=TS%(LF)-1:RETURN
1370 REM MENU
```

```
1380 PRINT C7$:PRINT
1390 PRINT HI$;"    GENERAL CONTEST LOGGER MENU          "
1400 PRINT HI$;"                                         "
1410 PRINT HI$;" '?' DISPLAYS THIS MENU                  "
1420 PRINT HI$;" 'QSY <BAND> <MODE>' CHANGES BANDS       "
1430 PRINT HI$;" 'DATE <MM/DD/YY>' CHANGES DATE          "
1440 PRINT HI$;" 'MODE <MODE>' CHANGES MODES             "
1450 PRINT HI$;" 'QRT' SAVES ALL CALLS AND EXITS         "
1455 PRINT HI$;" 'QRX' SAVES ALL CALLS AND STOPS TIMER   "
1460 PRINT HI$;" 'SAVE' SAVES ALL CALLS AND RESUMES      "
1470 PRINT HI$;"        (NOTE: ALLOW 2-3 MIN FOR SAVE)   "
1475 PRINT HI$;" 'RATE' SHOWS TOTAL TIME USED AND RATE   "
1480 PRINT HI$;" 'STAT' GIVES STATUS ON CURRENT BAND     "
1490 PRINT HI$;" 'SCORE' GIVES RUNNING TOTAL SCORE       "
1500 PRINT HI$;" 'EDIT <QSO #>' EDITS ANY LOGGED ENTRY   "
1510 PRINT HI$;" 'X' DELETES THE LAST ENTRY              "
1520 PRINT HI$;" '-' ABORTS CURRENT ENTRY                "
1530 PRINT HI$;" ENTRY FORMAT: 'CALL EXCHANGE <UTC>'     "
1540 PRINT HI$;" ENTER THE CALL ONLY FOR DUPE CHECK      "
1550 PRINT HI$;" ENTER CALL AND '?' TO LIST BANDS        "
1552 PRINT HI$;"        ON WHICH A STATION IS NEEDED     "
1555 PRINT HI$;"                                         "
1560 PRINT HI$;" <> DENOTES OPTIONAL ENTRIES             "
1570 PRINT LO$:RETURN
1580 REM QSY
1590 GOSUB 2400
1600 IF NL=1 THEN RETURN
1610 IF G$="" THEN PRINT:INPUT "WHAT BAND (LOG SHEET) DO YOU WANT ";G$
1620 FOR J=1 TO NL: IF G$=LN$(J) THEN 1660
1630 NEXT J
1640 PRINT "NO SUCH LOG. VALID LOGS ARE:";BL$
1650 FOR J=1 TO NL:PRINT TAB(5)LN$(J):NEXT J:G$="":GOTO 1610
1660 LG=J:IF HC$="Y" THEN GOSUB 2490
1670 GOSUB 2400:IF G$<>"" THEN GOSUB 1200
1680 RETURN
1690 REM STATUS
1700 PRINT C7$:PRINT:PRINT "DATE= ";DTE$;
1710 PRINT "  BAND= ";LN$(LG);
1720 PRINT "  MODE= ";MD$
1725 LF=LG:IF P2=0 THEN LF=1
1730 PRINT " SECTIONS WORKED= ";TS%(LF);
1740 PRINT "  CONTACTS= ";TC%(LG)
1750 PRINT:RETURN
1760 REM LIST SECTIONS
1770 PRINT C7$:PRINT:LF=LG:IF P2=0 THEN LF=1
1780 PRINT "------ MULTIPLIERS WORKED";:IF P2=1 THEN PRINT " ON ";LN$(LF);
1782 PRINT " ------"
1790 IF TS%(LF)=0 THEN RETURN
1800 TB=1
1810 FOR J=1 TO TS%(LF)
1820 PRINT TAB(TB)S$(J,LF);
1830 TB=TB+8:IF TB>38 THEN TB=1:PRINT
1840 NEXT J:PRINT:RETURN
1850 REM QRT
1860 CLOSE3
1870 OPEN4,8,4,FA$+",S,W":GOSUB 6000
1900 PRINT#4,HC$:PRINT#4,T:PRINT#4,Y$:PRINT#4,Z$:PRINT#4,NL
1901 PRINT#4,L%:PRINT#4,M1
1910 FOR J=1 TO NL:PRINT#4,LN$(J):NEXT J
1912 PRINT#4,ZA:PRINT#4,ET
1914 FOR J=1 TO ZA:PRINT#4,U1$(J):PRINT#4,U2$(J):PRINT#4,U3(J):NEXT J
1950 FOR J=1 TO NL:PRINT#4,QP%(J);",";TC%(J);",";TS%(J):NEXT J
1960 FOR J=1 TO NL:FOR K=1 TO TS%(J)
1970 PRINT#4,SC%(K,J);",";S$(K,J)
1980 NEXT K:NEXT J
1982 PRINT#4,P1$:PRINT#4,P2
1985 RETURN
1990 REM WRITE MULT SUMMARY TO #4
2000 PRINT#4,BL$:REM NEW PAGE
2010 PRINT#4,""SPC(27)"* MULTIPLIER SUMMARY *"
```

```
2015 LF=NL:IF P2=0 THEN LF=1
2020 FOR J=1 TO LF
2030 PRINT#4," ":PRINT#4," ":PRINT#4," "
2040 PRINT#4,"---------------------  MULTIPLIERS";
2041 IF P2=1 THEN PRINT#4," WORKED ON ";LN$(J);
2042 PRINT#4,"  ----------------------":PRINT#4," "
2050 FOR K=1 TO 5:PRINT#4,"  M'PLIER/QSOS ";:NEXT K:PRINT#4,""
2060 TB=9
2070 FOR K=1 TO TS%(J)
2080 PRINT#4,""SPC(9-LEN(S$(K,J)))S$(K,J);"/";SC%(K,J);
2082 PRINT#4,""SPC(5-LEN(STR$(SC%(K,J))));
2090 TB=TB+15:IF TB>74 THEN TB=9:PRINT#4,""
2100 NEXT K:PRINT#4," ":NEXT J
2110 REM WRITE SCORE TO #4
2120 PRINT#4,BL$:REM NEW PAGE
2130 PRINT#4,"-                SCORE SUMMARY":PRINT#4,"":PRINT#4,T1$
2140 IF HC$<>"Y" THEN 2150
2142 PRINT#130,"":PRINT#130," ";SPC(13)"SCORE SUMMARY"
2144 PRINT#130,"":PRINT#130,T1$
2150 X2=0:X3=0:X4=0
2160 FOR J=1 TO NL
2170 X4=X4+TC%(J):X1=TC%(J)*QP%(J):X2=X2+X1:X3=X3+TS%(J)
2180 PRINT#4,LN$(J);SPC(10-LEN(LN$(J)))TC%(J);
2181 PRINT#4,""SPC(9-LEN(STR$(TC%(J))))QP%(J);SPC(7-LEN(STR$(QP%(J))))X1;
2185 IF P2=1 THEN PRINT#4,"";SPC(10-LEN(STR$(X1)))TS%(J);
2186 PRINT#4,""
2188 IF HC$<>"Y" THEN 2200
2190 PRINT#130,LN$(J);SPC(11-LEN(LN$(J)))TC%(J);SPC(5)QP%(J);SPC(5)X1;
2192 IF P2=1 THEN PRINT#130,"      "TS%(J);
2194 PRINT#130,""
2200 NEXT J
2208 IF P2=0 THEN X3=TS%(1)
2210 PRINT#4,""T2$:PRINT#4,"TOTAL";SPC(6)X4;SPC(16-LEN(STR$(X4)))X2;
2212 PRINT#4,"";SPC(10-LEN(STR$(X2)))X3:PRINT#4,""
2220 IF HC$<>"Y" THEN 2230
2222 PRINT#130,T2$:PRINT#130,"TOTAL";SPC(5)X4;SPC(15-LEN(STR$(X4)))X2;
2224 PRINT#130," ";SPC(8-LEN(STR$(X2)))X3:PRINT#130,""
2230 PRINT#4,"*************************************"
2240 PRINT#4,"*                                   *"
2250 PRINT#4,"SCORE - ";X2;" X ";X3;SPC(10)" =";X2*X3
2260 PRINT#4,"*                                   *"
2270 PRINT#4,"*************************************"
2272 PRINT#4,"!"
2280 CLOSE4:GOSUB 6000
2281 GOSUB 4000
2282 FOR RN=1 TO L%:GOSUB 7000
2283 RP=1:GOSUB 7100:PRINT#3,C1%(RN)
2284 RP=9:GOSUB 7100:PRINT#3,C2%(RN)
2285 RP=17:GOSUB 7100:PRINT#3,P%(RN)
2286 GOSUB 6000:NEXT RN
2287 CLOSE3:GOSUB 6000:CLOSE15
2290 IF HC$="Y" THEN PRINT#130,"SCORE = ";X2*X3
2300 FOR J=1 TO 1500:PRINT;:NEXT J:CLOSE 130:END
2310 PRINT C7$:PRINT:REM PRINT TITLE
2320 PRINT:PRINT
2330 PRINT "  ";HI$;"  GENERAL CONTEST LOGGING PROGRAM  "
2335 PRINT "  ";HI$;"          BY KC6A AND N6NB          "
2340 PRINT "  ";HI$;" (C) 1983 BY HAYDEN BOOK CO., INC. "
2350 PRINT LO$:PRINT:RETURN
2360 REM INPUT SCAN ROUTINE
2370 GS$="":INPUT GS$:LL=LEN(GS$):IF LL=0 THEN 2430
2380 FOR J=1 TO LL:IF MID$(GS$,J,1)="-" THEN X1%=1:GOTO 650
2381 NEXT J
2390 LP=1
2400 IF LP>LL THEN 2430:REM 2ND ENTRY
2410 FOR LT=LP TO LL:IF MID$(GS$,LT,1)<>" " THEN 2440
2420 NEXT LT
2430 G$="":RETURN
2440 LP=LT
2450 FOR LT=LP TO LL:IF MID$(GS$,LT,1)=" " THEN 2480
```

```
2460 NEXT LT
2470 LT=LT+1
2480 G$=MID$(GS$,LP,LT-LP):LP=LT:RETURN
2490 PRINT#130,""
2500 PRINT#130,"     QSY TO: ";LN$(LG)
2510 RETURN
2520 LL=LEN(GS$):GOTO 2390
2530 REM RESUME
2540 OPEN4,8,4,FA$+",S,R":GOSUB 6000
2570 INPUT#4,HC$:INPUT#4,T:INPUT#4,Y$:INPUT#4,Z$:INPUT#4,NL
2571 INPUT#4,L%:INPUT#4,M1
2580 DIM LN$(NL),QP%(NL),TC%(NL),TS%(NL),SC%(100,NL)
2590 DIM C1%(M1),C2%(M1),P%(M1),S$(100,NL)
2600 FOR J=1 TO NL:INPUT#4,LN$(J):NEXT J
2602 INPUT#4,ZA:INPUT#4,ET
2604 FOR J=1 TO ZA:INPUT#4,U1$(J):INPUT#4,U2$(J):INPUT#4,U3(J):NEXT J
2650 FOR J=1 TO NL:INPUT#4,QP%(J),TC%(J),TS%(J):NEXT J
2660 FOR J=1 TO NL:FOR K=1 TO TS%(J)
2670 INPUT#4,SC%(K,J),S$(K,J)
2680 NEXT K:NEXT J
2682 INPUT#4,P1$:INPUT#4,P2
2690 CLOSE4:GOSUB 6000
2691 GOSUB 4000
2692 FOR RN=1 TO L%:GOSUB 7000
2693 RP=1:GOSUB7100:INPUT#3,C1%(RN)
2694 RP=9:GOSUB7100:INPUT#3,C2%(RN)
2695 RP=17:GOSUB7100:INPUT#3,P%(RN)
2696 NEXT RN:CLOSE3:GOSUB6000:RETURN
2700 REM EDIT A RECORD
2710 GOSUB 2400:IF G$="" THEN INPUT "WHAT QSO# DO YOU WANT TO EDIT ";G$
2720 X2=INT(VAL(G$))
2730 IF X2>0 THEN IF X2<T THEN 2750
2740 GOTO 670
2750 PRINT C7$:JB=8:IF P1$="Y" THEN JB=7
2760 RN=X2:GOSUB 7000
2761 RP=1:GOSUB 7100:INPUT#3,Y$(2):Y$(12)=Y$(2)
2762 RP=10:GOSUB 7100:INPUT#3,Y$(3):Y$(13)=Y$(3)
2763 RP=15:GOSUB 7100:INPUT#3,Y$(4):Y$(14)=Y$(4)
2764 RP=26:GOSUB 7100:INPUT#3,Y$(5):Y$(15)=Y$(5)
2765 RP=32:GOSUB 7100:INPUT#3,Y$(6):Y$(16)=Y$(6)
2766 RP=38:GOSUB 7100:INPUT#3,E1$
2767 GS$=E1$:GOSUB 2520:Y$(7)=G$:Y$(17)=G$
2768 Y$(8)=Y$(7):IF P1$<>"Y" THEN GOSUB 2400:Y$(8)=G$
2769 Y$(18)=Y$(8)
2770 GOSUB 3080
2780 PRINT:A$="":INPUT "CHANGE THIS RECORD (Y/N) ";A$
2790 IF A$<>"Y" THEN PRINT C7$:GOTO 450
2800 LH%=0
2810 FOR J=1 TO NL:IF Y$(5)=LN$(J) THEN 2830
2820 NEXT J
2830 LX=LG:LY=J:LK=LY
2840 PRINT:PRINT TAB(12)"INSTRUCTIONS":PRINT
2850 PRINT "HIT <CR> TO LEAVE AN ITEM UNCHANGED."
2860 PRINT "TO CHANGE, ENTER NEW VALUE <CR>."
2870 PRINT "TO DELETE THIS ENTRY, CHANGE CALL TO '*'."
2878 PRINT
2880 A$="":PRINT TAB(5)"DATE:";TAB(14)Y$(2);:INPUT A$:IF A$<>"" THEN Y$(2)=A$
2890 PRINT TAB(5)"TIME:";TAB(14)Y$(3);:A$="":INPUT A$:IF A$<>""THEN Y$(3)=A$
2900 PRINT TAB(5)"CALL:";TAB(14)Y$(4);:A$="":INPUT A$:IF A$<>""THEN Y$(4)=A$
2910 PRINT TAB(5)"BAND:";TAB(14)Y$(5);:A$="":INPUT A$:IF A$<>"" THEN Y$(5)=A$
2920 PRINT TAB(5)"MODE:";TAB(14)Y$(6);:A$="":INPUT A$:IF A$<>""THEN Y$(6)=A$
2930 IF P1$<>"Y" THEN PRINT TAB(5)"EXCHANGE:";TAB(14)Y$(7);
2931 IF P1$<>"Y"THEN A$="":INPUT A$:IF A$<>"" THEN Y$(7)=A$
2935 PRINT TAB(5)"M'PLIER:";TAB(14)Y$(8);:A$="":INPUT A$:IF A$<>"" THEN Y$(8)=A$
2940 PRINT:A$="":INPUT "OK (Y/N) ";A$:X1%=4
2950 IF A$<>"Y" THEN PRINT C7$:GOSUB 650:GOTO 450
2960 IF Y$(5)<>Y$(15) THEN GOSUB 3240:GOTO 3000:REM BAND CHANGE
2970 LG=LK
2980 IF Y$(4)<>Y$(14) THEN GOSUB 3160:GOSUB 3190:REM CALL CHANGE
2990 IF Y$(8)<>Y$(18) THEN GOSUB 3320:GOSUB 3350:REM EXCHANGE CHANGE
```

```
3000 LG=LX
3010 ON X1% GOTO 590,3370,3370,3020,3020,3380
3020 RP=1:GOSUB 7100:PRINT#3,Y$(2)
3021 RP=10:GOSUB 7100:PRINT#3,Y$(3)
3022 RP=15:GOSUB 7100:PRINT#3,Y$(4)
3023 RP=26:GOSUB 7100:PRINT#3,Y$(5)
3024 RP=32:GOSUB 7100:PRINT#3,Y$(6)
3032 IF P1$="Y" THEN Y$(7)=""
3040 RP=38:GOSUB 7100:PRINT#3,Y$(7)+" "+Y$(8)
3050 IF HC$<>"Y" THEN 3070
3060 PRINT#130,"  ** EDIT ON REC #";X2;": ";D1$;" ";TM$;" ";KL$;
3061 PRINT#130," ";B1$;" ";M1$;" ";E1$
3070 GOTO 450
3080 PRINT:PRINT  TAB(5)"DATE:";TAB(14)Y$(2)
3090 PRINT TAB(5)"TIME:";TAB(14)Y$(3)
3100 PRINT TAB(5)"CALL:";TAB(14)Y$(4)
3110 PRINT TAB(5)"BAND:";TAB(14)Y$(5)
3120 PRINT TAB(5)"MODE:";TAB(14)Y$(6)
3130 IF P1$<>"Y" THEN PRINT TAB(5)"EXCHANGE:";TAB(14)Y$(7)
3135 PRINT TAB(5)"M'PLIER:";TAB(14)Y$(8)
3140 RETURN
3150 REM CALL CHANGED
3160 IF Y$(4)="*" THEN Y$(8)="*":RETURN
3170 CS$=Y$(4):GOSUB 3400:IF X1%<4 THEN RETURN:REM CHECK IF NEW CALL IS DUPE
3180 GOSUB 990:TC%(LG)=TC%(LG)+1:RETURN
3190 IF X1%<4 THEN RETURN
3195 IF Y$(14)="*" THEN RETURN
3200 CS$=Y$(14)
3210 GOSUB 3400:C1%(H%)=-32001:C2%(H%)=-32000:REM DELETE OLD CALL
3220 TC%(LG)=TC%(LG)-1:X1%=4:RETURN
3230 REM CHANGE BAND
3240 FOR LK=1 TO NL:IF Y$(5)=LN$(LK) THEN 3270
3250 NEXT LK
3260 X1%=6:RETURN
3270 LG=LK:GOSUB 3160:IF X1%<4 THEN RETURN
3280 LG=LY:GOSUB 3190
3290 GOSUB 3320:LG=LK:GOSUB 3350:RETURN
3300 LG=LY:GOSUB 3350:RETURN
3310 REM CHANGE EXCH
3320 IF Y$(18)="*" THEN RETURN
3330 SK$=Y$(18)
3340 GOTO 1310:REM DELETE OLD SECTION
3350 IF Y$(8)="*" THEN RETURN
3360 SK$=Y$(8):GOSUB 1020:RETURN:REM ADD NEW SECTION
3370 PRINT BL$;"INVALID CALL SIGN":GOTO 450
3380 PRINT BL$;"INVALID BAND":GOTO 450
3390 REM DUPE CHECKER: X1%=1=DUPE: 2=QUERY: 3=INVALID CALL: 4,5=OK
3400 X1%=4:E(5)=0:E(6)=0:K=1:IF LEN(CS$)>10 THEN X1%=3:RETURN
3410 IF RIGHT$(CS$,1)="?" THEN CS$=LEFT$(CS$,LEN(CS$)-1):X1%=2:CT$=CS$
3420 IF LEN(CS$)<4 THEN X1%=3:RETURN
3430 IF MID$(CS$,3,1)<"A" THEN 3460
3440 IF MID$(CS$,2,1)=>"A" THEN X1%=3:RETURN
3450 CS$=" "+CS$:K=2:E(1)=0
3460 FOR J=K TO LEN(CS$)
3470 W$=MID$(CS$,J,1):IF W$="/" THEN 3520
3480 E(J)=ASC(W$)-AA:IF E(J)>26 THEN X1%=3:RETURN
3490 IF W$<"0" THEN X1%=3:RETURN
3500 IF E(J)<1 THEN E(J)=E(J)+A0:IF E(J)>36 THEN X1%=3:RETURN
3510 NEXT J
3520 IF J>7 THEN X1%=3:RETURN
3530 IF J-K<4 THEN X1%=3:RETURN
3540 FOR J=4 TO 6:IF E(J)>26 THEN X1%=3:RETURN
3550 NEXT J
3560 E1%=-32000+E(1)*1369+E(2)*37+E(3)
3570 E2%=-32000+E(4)*1369+E(5)*37+E(6)
3580 IF X1%=2 THEN RETURN
3590 H%=E(4)+LG*26-26
3600 REM SEARCH DUPE STRING
3610 IF C1%(H%)=-32000 THEN X1%=4:RETURN
3620 IF C2%(H%)<>E2% THEN 3640
3630 IF C1%(H%)=E1% THEN X1%=1:RETURN
```

```
3640 IF P%(H%)=0 THEN X1%=5:RETURN
3650 H%=P%(H%):GOTO 3620
3660 REM SCORE
3670 PRINT:PRINT TAB(10)"SCORE SUMMARY":PRINT
3680 PRINT T1$
3690 X2=0:X3=0:X4=0
3700 FOR J=1 TO NL
3710 X4=X4+TC%(J):X1=TC%(J)*QP%(J):X2=X2+X1:X3=X3+TS%(J)
3720 PRINT LN$(J);TAB(11)TC%(J);TAB(20)QP%(J);TAB(27)X1;
3722 IF P2=1 THEN PRINT TAB(37)TS%(J);
3724 PRINT
3730 NEXT J
3740 PRINT T2$
3750 PRINT "TOTAL";TAB(11)X4;TAB(27)X2;TAB(37)X3
3760 PRINT:PRINT TAB(10)"SCORE =";X3*X2
3770 PRINT:RETURN
3780 IF A$="R" THEN 3800
3790 OPEN3,8,3,FC$+",L,"+CHR$(53):GOSUB 6000:RETURN
3800 OPEN3,8,3,FC$:GOSUB 6000:RETURN
3900 IF RC$<>"Y" THEN 3920
3910 GOSUB 1162:U2$(ZA)=UTC$:GOTO 3930
3920 PRINT:INPUT "ENTER UTC TIME";U2$(ZA)
3930 UX=VAL(U1$(ZA)):UY=VAL(U2$(ZA))
3932 U7=INT(UX/100):U4=((UX/100)-U7)/.6:U7=U7+U4
3940 U5=INT(UY/100):U6=((UY/100)-U5)/.6:U5=U5+U6
3950 IF U5<U7 THEN U5=U5+24
3960 E7=ET:ET=ET+U5-U7:U3(ZA)=T:IF ET=0 THEN GOTO 3990
3970 PRINT:PRINT "TIME USED SO FAR: ";
3972 PRINT INT(ET);" HRS - ";INT((ET-INT(ET))*60+.5);" MINS"
3980 PRINT:PRINT "OVERALL RATE:  ";INT((T-1)/ET+.5);" QSOS PER HOUR"
3990 RETURN
4000 IF A$="R" THEN 4020
4010 OPEN3,8,3,FB$+",L,"+CHR$(23):GOSUB 6000:RETURN
4020 OPEN3,8,3,FB$:GOSUB 6000:RETURN
6000 INPUT#15,EA,EB$,EC,ED:IF EA<20 THEN RETURN:REM READ ERROR CHANNEL
6010 IF EA=50 THEN RETURN
6020 IF EA=63 THEN CLOSE4:OPEN4,8,4,"@"+FA$+",S,W":RETURN
6030 PRINT EA;EB$,EC;ED:STOP:RETURN
7000 RH=INT(RN/256):RL=RN-256*RH:RETURN
7100 PRINT#15,"P"CHR$(3)CHR$(RL)CHR$(RH)CHR$(RP)
7110 GOTO 6000
```

OPERATING PROCEDURES FOR THE CONTEST LOGGERS

Here's a step-by-step description of how the GENERAL PURPOSE CONTEST LOGGER works. The special features of the FIELD DAY LOGGER and SWEEPSTAKES LOGGER—and operating hints for first-time users of a computer for contest logging—are also summarized.

The Initial Setup

1. Plan to have the computer set up for the contest well before the starting time. You'll need to tell the LOGGER a number of details of the rules and scoring system, so that the dupechecking and on-line scoring features of the program will both work correctly.

2. Use a new disk that contains the LOGGER and nothing else. The disk must not have a lot of other files on it, since a substantial amount of disk space is needed to store the data files generated by the LOGGER. When you initiate BASIC, some computers (e.g., the TRS-80) require

you to specify the number and type of files when you load BASIC. Specify three data files of variable length ("3V" on a TRS-80). This happens automatically on most other computers: you simply load and run the program.

3. The LOGGER first asks if you have a real-time clock. If you do—and you want to use it—enter "Y" for yes. If you don't, or if this is a post-contest retyping of handwritten logs, type "N." If you have a clock, by all means use it. Entering the time for every QSO is a real nuisance. But make sure the clock is accurate before you start out. You wouldn't want to be accused of "rubber clocking."

4. Next, the program asks if you're starting a new contest or resuming after a break. If you type "N" (for new), you'll create new data files for each band, and this will wipe out any data files that were already on the disk. If you say "R," the existing files will be read and you can pick up where you left off. The LOGGER will simply ask which band you want to operate in that case.

5. You'll now have to decide whether you want a hard copy printed while you're running the contest. A hard copy is a useful form of backup, just in case the unthinkable happens and your battery power fails immediately after the commercial power goes down (Murphy's Law says that can happen, you know). What, you don't have battery backup???

   However, you'll be printing a neater set of logs for submission after the contest, using the LOGPRINT program, so you don't really need a hard copy while the contest is going on except for backup.

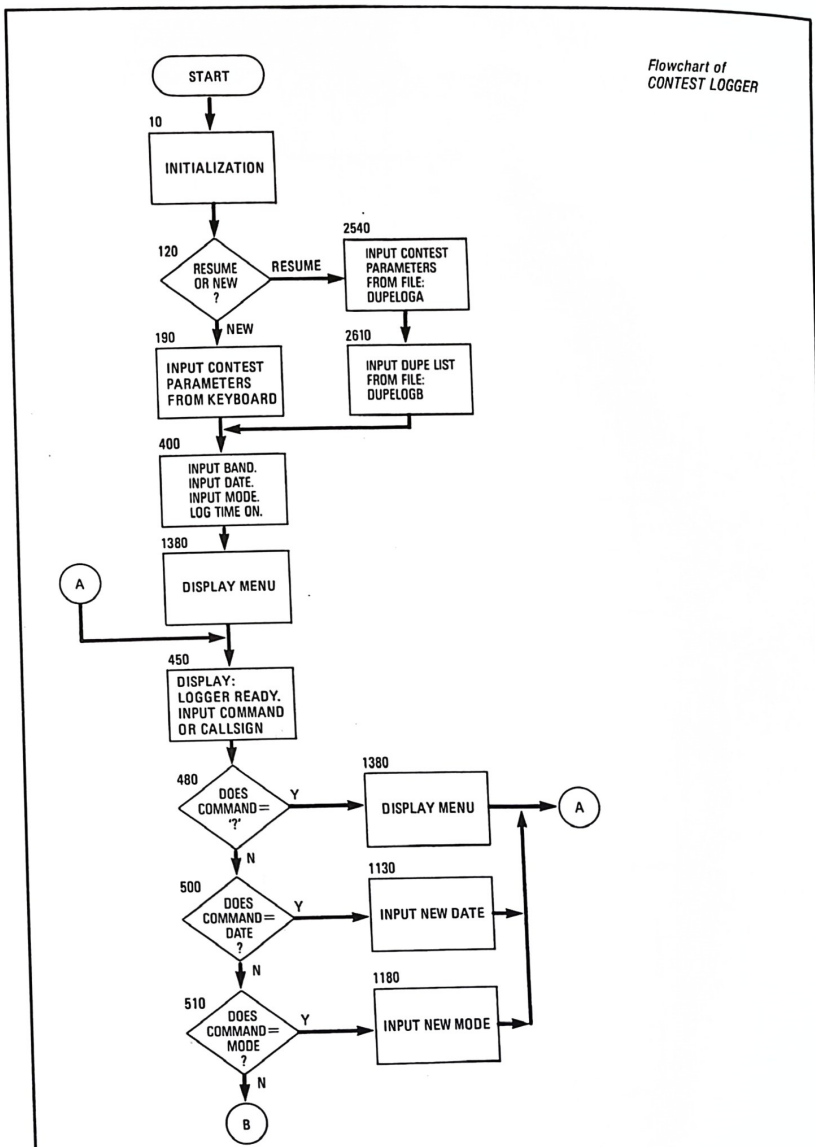
6. At this point, you'll enter the contest name, which is just a formality to ensure that the log sheets are correctly labeled.

Bands, Multipliers and QSO Points

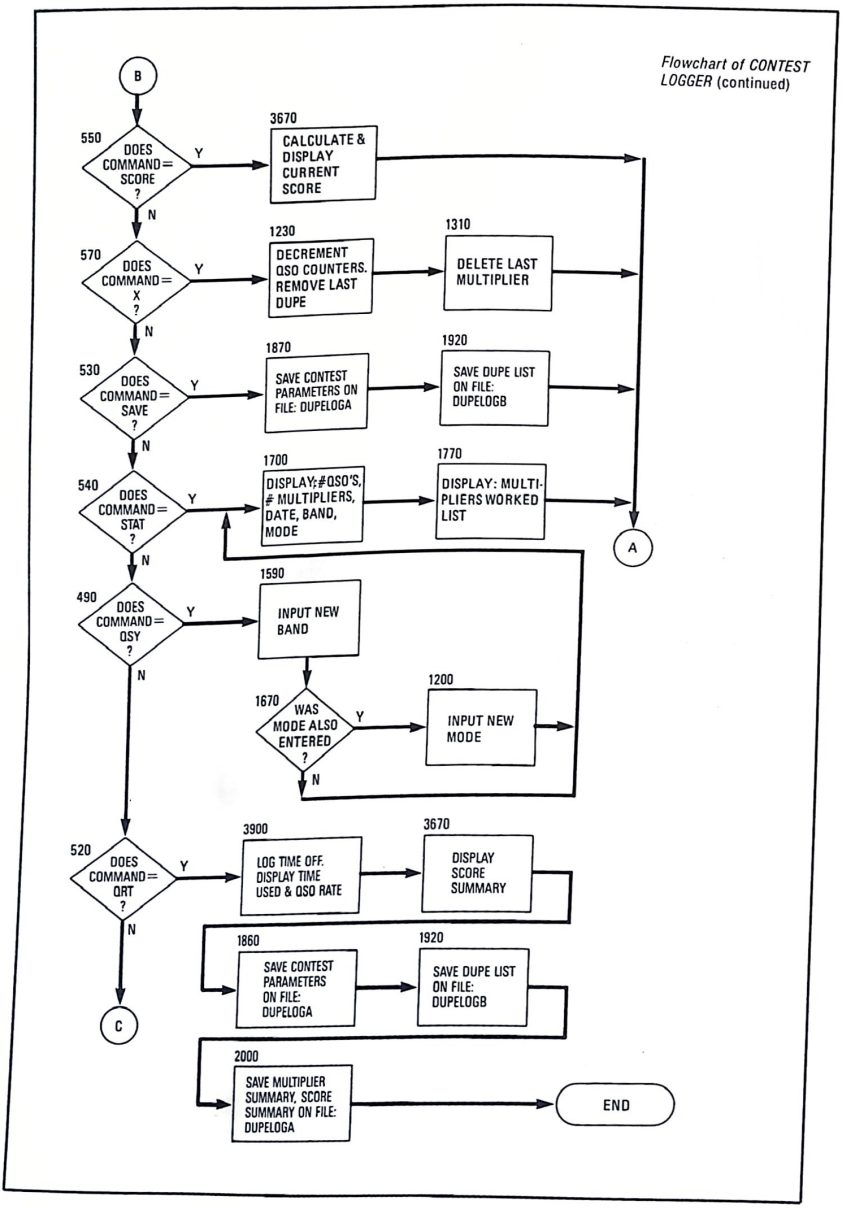
7. If you're starting a new contest, the Logger will ask how many separate log sheets (i.e., bands) you want. You may specify one to ten. If this is a contest in which you can only work a station once regardless of band, enter "1" and use the MODE command (explained shortly) to log band changes. That way, the dupechecking feature will work correctly when you change bands, preventing you from duping stations that you worked on a different band.

   If, however, this is a typical contest in which you can work a station once per band (with repeat contacts on other bands permitted), just enter the maximum number of bands on which you might operate. Be sure to allow for enough bands. If you don't use a band for which you've created a data file, there's no problem. But if you decide you want to add another band midway through the contest, you're out of luck.

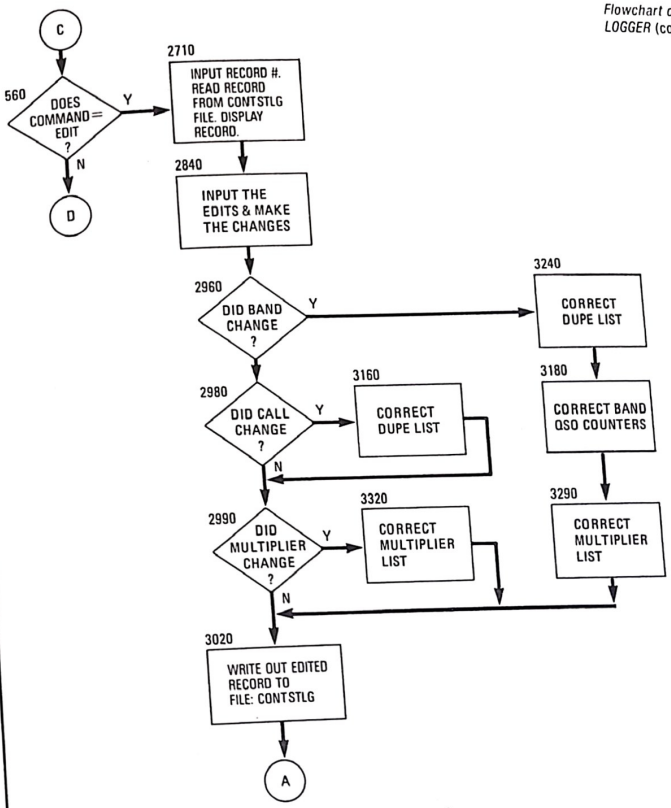
8. Next, the LOGGER will ask you to enter the name of each band, along with the point-value for QSOs on that band. In some contests, it's



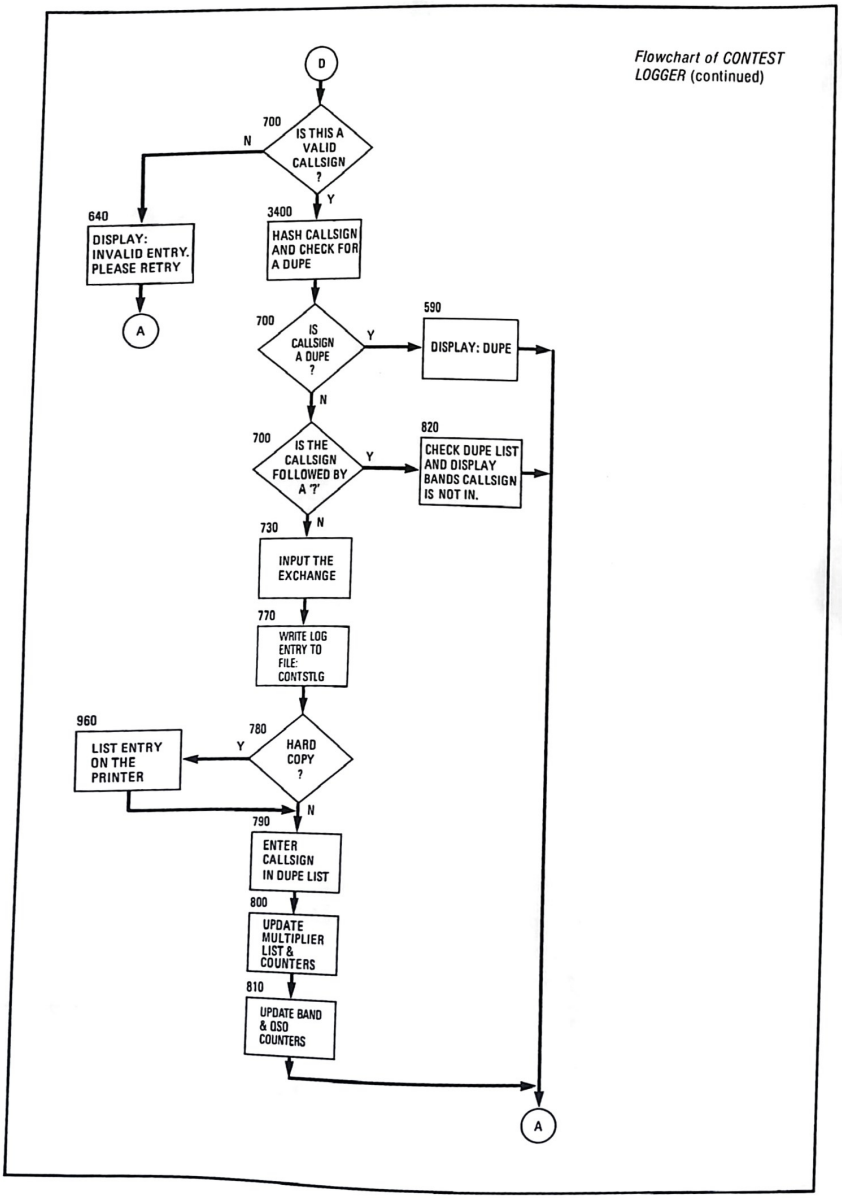
Flowchart of CONTEST LOGGER



Flowchart of CONTEST LOGGER (continued)



Flowchart of CONTEST LOGGER (continued)



Flowchart of CONTEST LOGGER (continued)

always the same, but it isn't in most VHF-UHF contests. This feature allows you to adjust the scoring for the unique rules of many different contests.

You could also treat the CW and phone segments of a band as different bands and specify different point values for CW and voice QSOs, as you would in the ARRL 10-Meter Contest. Since you can work the same station on both voice and CW in that contest, the dupechecking feature will work correctly if you set up 10CW and 10PH (for phone) as separate bands. Check the official contest rules for the point value in the particular contest you're going to operate.

9. After you name your bands and specify QSO point values, the LOGGER will create a data file for each band.

10. Now comes one of the key questions. The LOGGER will ask whether there are separate multipliers on each band or if you can only count a multiplier once during the entire contest. In most DX and VHF contests, there are band multipliers, but in VHF Sweepstakes, for example, that's not true: the multipliers only count once, even though you can work each station again for credit on additional bands. If there are separate band multipliers in the contest you're setting the LOGGER up for, answer this question with a "Y" (yes).

    Note that you have a wide variety of options here. By treating the entire contest as one "band" and using the "MODE" command to log band changes, you can run a contest in which neither QSOs nor multipliers can be repeated on additional bands. On the other hand, by specifying multiple bands with separate multipliers for each, you can set the LOGGER up for contests where both QSOs and multipliers can be repeated on additional bands. And if you specify multiple bands but say "N" (no) in response to this question, you can handle contests that lack band-multipliers but allow QSOs to be repeated on additional bands for credit. You have your choice of all three combinations of band-by-band scoring credit.

11. Here's another question that is central to the LOGGER's versatility. You will be asked whether the entire exchange is the multiplier (as opposed to there being a nonscoring part such as a signal report, ARRL appointment, serial number, or other required additional information). If the entire exchange is the multiplier (as it is in some VHF contests), answer "Y." If you want a two-part exchange, answer "N." If you answer no to this question, you will be expected to enter the nonscoring part of the exchange first, then hit the space bar, and finally type the multiplier. (Example: "599 LA".)

    The LOGGER keeps track of the multipliers for you. You can print a list of the multipliers you've worked on any band, and you'll be told each time you work a new multiplier! But there's one crucial requirement: you must use the same designation for each multiplier every time. Suggestion: post a list of standard multipliers on the wall, and use them at all times. If

you enter "NLI" for the New York–Long Island section once, and then say "NYCLI", the logger will give you credit for two different multipliers.

And if you forget to separate the nonscoring part of the exchange from the multiplier, you'll also have problems. Don't forget the all-important space between the report (or other nonscoring information) and the multiplier.

Starting a Contest

12. After you've set up the LOGGER for the rules of the particular contest, you're almost ready to begin operating. Now enter the date and indicate which band you want to operate. Finally, enter the mode, which can be up to three characters. As indicated earlier, the mode command can also be used to log band changes in a contest that only permits you to work stations once for credit, regardless of band. In such a contest you only specify one band for dupechecking purposes.

13. Now you're ready to go. You'll be asked to type a carriage return to mark the starting time when the contest begins and you're ready to start operating. If you're not using a real-time clock, you will also be asked to enter the starting time. At this point the main menu appears and you're ready to start operating. There will be a prompt for the first QSO ("COMMAND OR CALL?").

You can begin entering calls, exchanges, and (if you're not using a real-time clock) the UTC time. There's a short delay to initialize the dupechecking feature on the first QSO, but after that the LOGGER will do the dupechecking and be ready to accept each new exchange within a second or two. The duping information is stored in RAM to assure rapid responses even with a log-full of calls in memory.

In entering each QSO, you have two options:

A. You can just enter the call sign and wait for a dupecheck; or

B. You can enter the call sign, hit the space bar, enter the nonmultiplier part of the exchange (if any), hit the space again, and enter the multiplier—all on one line. If you're not using a real-time clock, you can also enter the time on that line, with a space separating it from the exchange. If you omit part of this information, you'll be prompted for the rest—provided you enter everything in the right order and separate the items with spaces. If you enter something in the wrong order or notice a typing error after hitting a carriage return, simply delete the QSO and try again. You can abort the current entry by typing a minus sign, or delete the contact just logged by typing "X."

If you enter everything at once, it's faster, but that means you don't get a dupecheck on the station before completing the contact. If it is a dupe, of course, the information won't be stored, and the "IT'S-A-DUPE" message will appear.

You can skip the UTC time entry by hitting another carriage return when the "UTC?" prompt appears. This feature was included because some operators prefer not to enter the time for every single

contact. If you have a real-time clock, of course, you don't have to worry about this—it happens automatically.

Suggestions for Computerized Operating

14. Suppose you're running a multi-operator contest, and you have someone you don't know doing your logging and dupechecking. It takes a while to get to know this person's style and capabilities, right?

When you first use a computer as your second operator, there is also an adjustment period. But you'll probably discover that a computer can be at least as valuable as having a second operator at your side—and it won't cost you your "single operator" status.

Once the learning period is over, you may be amazed at how convenient computerized contesting can be. Compared to the old way of juggling logs and big, clumsy dupesheets, it's terrific. And the versatility it gives you in terms of seeing what multipliers you still need on any band at any time! There's just nothing like it for efficiency, not to mention the dupes you avoid and the time and trouble you save after the contest.

Some people have difficulty getting used to the idea of doing the original entry of data on a keyboard. "Don't you write it down first and then type it later?" they ask.

The answer is no—just as the answer would be no if the question were, "Didn't you write the manuscript of this book in longhand first, and type it later?" Many computer users learn to type much faster than they can write with a pencil. After a while, it becomes a real nuisance to write anything out in longhand.

Yes, we do log directly on the keyboard.

If you are just learning how to log on a keyboard, we would suggest that you practice on a contest with a short and simple exchange before you tackle something like ARRL Sweepstakes, which has a long and cumbersome exchange. If all you have to type is the call and a short exchange, computerized logging procedures are easier to learn.

If you're an experienced contest operator, you've probably discovered the contester's trick of using headphones, a boom mike, and a foot switch so you have both hands free for logging, duping, and equipment adjustments. When logging a contest on a computer, this technique is recommended. The ideal logging computer is a small, low-profile one, so the keyboard can sit right in front of the rigs—or be pushed aside quickly if necessary. Some Apple users place the main transceiver *on top* of the computer and put the video monitor on a shelf, back out of the way! In terms of size, a computer with a small detachable keyboard is excellent for contest logging.

There are basically two ways to operate a contest: you can "hunt and pounce," or you can "run." If you hunt and pounce (i.e., tune around looking for stations to call), the dupechecking feature of the logger programs become especially important. You don't want to look

ridiculous by repeatedly calling stations you've already worked. If you're hunting and pouncing, enter the call as soon as you find a likely station to work. That gets you a dupecheck. Then if you make the contact, type the rest of the information on one line, using spaces to separate the portions of the data. If you don't make contact, just abort the entry and go on.

If you're running (i.e., sitting on a frequency and working stations rapidly, one after another), enter the call as soon as you're reasonably sure you have it right, and hit a return. The logger will do your dupecheck while you're sending your exchange. In most cases you'll know if it's a dupe before you can acknowledge the caller and say something like "five-nine Maryland." Then type in the exchange as the calling station gives it, again using a single line for everything.

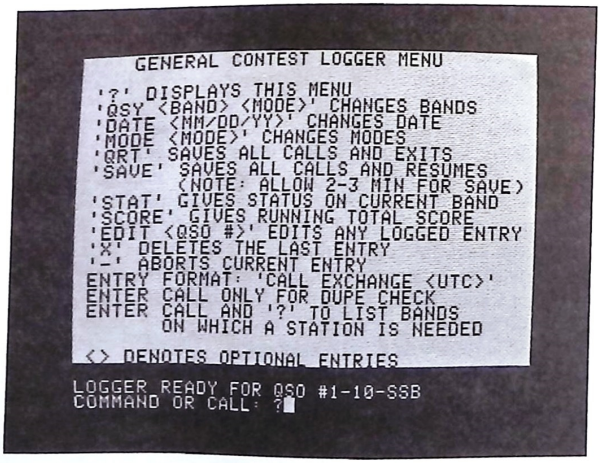
We use a scratch pad only when a station is so weak that we're unable to copy the information in proper order. Normally, we do it all electronically. Airline reservation agents type everything directly into their computers as they hear it over their headphones. Why shouldn't contest operators be at least that efficient?

Menu Options

Let's assume the contest is well underway now. It's time to look at the options on the main menu.

15. If you want to review the menu at any time, just type a "?" at the prompt. The menu will be immediately redisplayed.

16. After you have logged a number of contacts, try some of the menu options. "SAVE" will allow you to write all logged QSOs to disk, while


GENERAL CONTEST LOGGER MENU
'?' DISPLAYS THIS MENU
'QSY <BAND> <MODE>' CHANGES BANDS
'DATE <MM/DD/YY>' CHANGES DATE
'MODE <MODE>' CHANGES MODES
'QRT' SAVES ALL CALLS AND EXITS
'SAVE' SAVES ALL CALLS AND RESUMES
(NOTE: ALLOW 2-3 MIN FOR SAVE)
'STAT' GIVES STATUS ON CURRENT BAND
'SCORE' GIVES RUNNING TOTAL SCORE
'EDIT <QSO #>' EDITS ANY LOGGED ENTRY
'X' DELETES THE LAST ENTRY
'_' ABORTS CURRENT ENTRY
ENTRY FORMAT: 'CALL EXCHANGE <UTC>'
ENTER CALL ONLY FOR DUPE CHECK
ENTER CALL AND '?' TO LIST BANDS
ON WHICH A STATION IS NEEDED
<> DENOTES OPTIONAL ENTRIES
LOGGER READY FOR QSO #1-10-SSB
COMMAND OR CALL: ?

The CONTEST LOGGER offers the operator many options, as this menu shows. The LOGGER will give a quick status report on a band-by-band basis. It includes all of the features of the DUPECHECKER program, and it keeps track of such things as the total operating time and QSO rate.

"QRT" will terminate the logging session. In either case, everything in RAM must be written in a disk file, which will take several minutes, depending on the speed of your disk drive system.

Perhaps you want to take a break, but don't want to shut everything down. If you type "SAVE," the clock will keep running and your break time will count against your operating period. If you type "QRT," you'll be logged off the air and the break time won't be counted as operating time. But then you'll have to rerun the program to resume the contest.

That's where the "QRX" function comes in. It's halfway between "SAVE" and "QRT." It saves all contest data and stops the clock, but it doesn't terminate the session. When you type "QRX," the total operating time you have used so far will be displayed, along with your average QSO rate (in contacts per hour). To resume, all you have to do is indicate your new starting time (or enter a carriage return to log the new starting time if you're using a real-time clock). The LOGGER has prompts on the screen to lead you through these steps.

A closely related command is "RATE." This command doesn't interrupt your operation to write anything to disk, but it does display your QSO rate per hour and the total operating time you have used so far.

17. To change bands, just type "QSY." You'll be immediately logged onto the new band, and you'll get a quick summary of where you stand on this new band. You can also type "QSY 40" or the like, putting the QSY command and the new band all on one line, separated by a space. This will save time.

If, however, you're treating the entire contest as one "band," you can log a band change by typing "MODE" and specifying the new band as a new "mode." The final printed log shows both the "band" and the "mode" for each QSO, so this poses no problems in logging.

There are several commands, incidentally, that permit you to enter the data with the command on one line to save time. In addition to a command like "QSY 40," the logger will accept a command such as "MODE SSB" or "DATE 12/13/84."

It is also possible to change both the band and mode with a single QSY command as follows: "QSY 40 SSB". If there are more characters following an acceptable band name (and separated by a space), the contest logger programs assume the additional item is a mode change.

18. If you type "STAT," you get a rundown on the current band: the LOGGER will display your QSO total and it will list all of the multipliers you've worked on that band.

19. If you want a rundown on all bands, type "SCORE." Then your overall totals will be displayed. This is especially useful as the contest nears its conclusion. You can quickly see if you're closing in on a new record or

beating the competition across town (provided, of course, that the competition is also computerized, and is willing to give you an honest report of what his "SCORE" command is telling him).

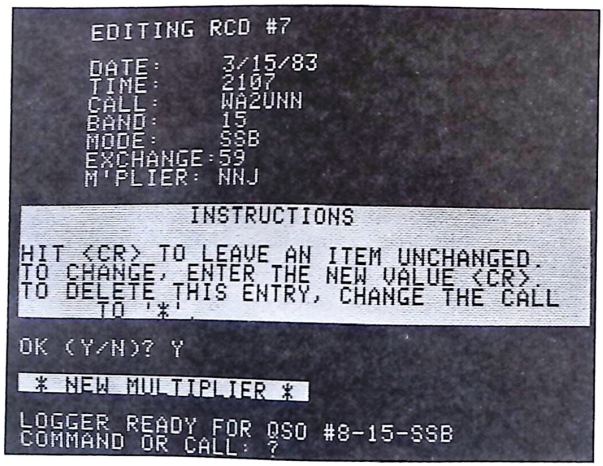
20. The LOGGER has all of the dupechecking capabilities of DUPECHECKER, including the multiband dupecheck. To see whether you still need a particular station on any band, type the call sign followed by "?" You'll get a list of all bands on which you need the station, with the current band flagged—if you need the station on this band.

Editing Features

21. Sometimes during a contest you discover that an earlier log entry is incorrect. Perhaps band conditions improve and you discover you've miscopied someone's call or multiplier. The "EDIT" command allows you to revise any earlier entry, either during the contest or afterward. You can change any of the particulars, including the call, multiplier, time, date, band, and mode.

After you type EDIT, the LOGGER will ask you for the "record number" of the QSO you want to change. If you prefer, you can enter the record number with EDIT (as in "EDIT 307"). The record number appears in the left-hand column of the hard-copy printout, and at the right side of the final log that is printed after the contest. If you can't locate the record number immediately, write down the new information and make the correction after the contest.

22. The EDIT command has its own menu. Editing a QSO is very easy. You'll see all of the specific information about the QSO displayed, and you can either change each item or type a return to leave it as is.



The three LOGGER programs provide full editing capabilities. Any aspect of a previous contact may be edited at any time, and the program will keep track of the effect the changes may have on the total score. If a change results in a new multiplier (or in the deletion of a multiplier), the LOGGER will take appropriate measures to correct the score.

When you're finished editing, you get one more chance to confirm that everything is correct, or you can abort the entire EDIT, leaving the QSO as it was before.

23. In addition to the EDIT command, there are two other ways to change entries, as mentioned earlier. Both are forms of what might be called a "banzai" command. Suppose you get a QSO logged and then discover the other guy didn't copy you. After much effort, you give up on the QSO. But it's in the log. Any time you want to delete the QSO that you have just logged, type "X". And if you give up on a QSO before you're finished logging it, type "–" (the minus sign) to abort the entry.

Printing the Logs and Dupesheets

24. When it's all over, don't forget to use the "QRT" command to save everything—and then get some sleep! Or if you prefer, you can show up on a 75-meter net right after the contest knowing your exact score and multiplier total—with all dupes removed. You may not want to *admit* your exact score at that point—there's something of a poker game that goes on during those post-contest brag sessions, isn't there? But you will know where you stand, in any case.

When you've recuperated from the big battle, you can run the LOGPRINT program to print a neat set of contest logs. LOGPRINT will read the data files generated by the LOGGER, and it will print out the results either in the usual time-sequential way or by bands (useful in certain specialty contests). In addition to a neat set of logs, you'll get detailed scoring and multiplier summaries. And if, heaven forbid, you see an error in the logs, you can always use the "RESUME" command to re-enter the logger program, edit out the error, and reprint the logs—painlessly!

As the text of this chapter explains, when you re-enter the program after the contest to edit out errors, you should not use a real-time clock even if you have one. Instead, select the option of entering the time manually. Whenever you RESUME the contest (even if it's just to edit), a time on and time off will be written into the data files. To

Sample printout of dupesheet from CONTEST LOGGER.

```
9/8/79                          N6NB                          PAGE1
                        SEPT 79 VHF CONTEST
                          DUPESHEET 144

                               -0-

WOFY        WDOFZP      NOIS        KOJXI

                               -1-

 NlABA       KlBKK      WAlEHL      WBlGRH      WlJSM       KlRT       WAlVUW
WAlABL       KAlBRD     WBlEIH       KlGSR      KlJX        WlRVQ      WAlWDL
 WlAC        KAlBRL      WlEJ        KlGVM     AElK        VElSJ       KlWGN
 NlAF        KAlBU       KlEM       WBlGXM      WlKCR       KlSND      WAlWRM
 KlAIK       WBlBVR     WBlEUC      WBlGXN      WlKUL       KlSU        WlWS
 WlAIM       WBlBZJ     WBlEUY       WlGXT      KlLIQ      WAlSVR      WAlWVY
```

N1AIS
N1AIU
WB1ALW
N1AMB
WB1ANT
N1APE
W1AQO
N1ASG
VE1ASJ
K1AU
WB1AUV
WA1AYS
W1BAT
KA1BBK
W1BJ
WB1CAG
K1CFR
WB1CJT
WB1CNQ
K1CPG
WA1CVN
WB1CZE
K1DAT
KA1DLW
WB1DNM
W1DO
K1DPP
KA1DQJ
K1DS
WA1EDX
W1FJH
WB1FKF
WB1FLY
K1FO
W1FVY
WB1FZI
K1GAO
WA1GDR
WB1GDU
WA1GQE
K1GQH
K1GYT
W1HAD
WB1HAE
W1HHE
W1HKY
W1HNP
WA1HNQ
W1HNZ
N1HR
H1HZN
AC1J
W1JEO
W1JP
K1JPP
W1JSH
K1LPS
WA1MAG
WB1MAO
K1MON
WA1MTI
W1MWF
WA1NGR
K1NJ
K1NKR
W1OJ
W1OKH
W1ONK
K1PAI
W1PP
WA1RLO
AF1T
K1THP
K1TK
K1TOZ
W1TS
WA1TTL
W1TZZ
WA1UQC
K1UR
VE1UT
WA1UVW
WA1VEI
K1VIJ
K1VMI
WA1VQP
K1WW
W1XA
W1XM
K1XR
W1XX
WA1YEC
WA1ZAM
WA1ZNE
WA1ZNT

9/8/79

N6NB
SEPT 79 VHF CONTEST
DUPESHEET 144

PAGE2

-2-

W2AAV
WA2AEH
W2AET
WA2AFE
WA2AHC
WB2ALQ
WA2ALV
WB2AMK
WA2ANZ
N2AQJ
W2ARQ
N2AU
W2AV
KA2AVA
WA2AWX
KA2AYQ
N2AZC
VE2BBK
WB2BDX
N2BFJ
K2BOG
WA2BPE
WA2BQL
K2BQZ
K2BWR
AF2C
K2CBA
KA2CDJ
WB2CEP
WB2CLU
W2CNF
WA2CQC
W2CRS
WB2CUT
K2CV
WA2CWA
W2CXM
W2DBQ
KA2DII
WB2DIN
WA2DJN
WA2DKB
WB2DNE
WA2DPU
W2DRB
KA2DTE
W2DUC
W2DYY
AK2E
WA2ECI
WB2EEC
W2ELM
WA2EMB
KA2ESB
KA2FAY
VE2FCX
WB2FFY
VE2FMF
WB2FPC
VE2FRT
WA2FUZ
W2FWA
AD2G
K2GE
WB2GEX
WA2GFP
WB2GIU
WA2GJA
K2GK
WA2GLJ
W2GN
WA2GPB
WA2GSX
WB2GXF
WB2HHH
W2IEC
WB2IEY
WB2IFN
WA2IKO
W2IKP
W2ISB
N2IY
WB2IYS
K2JIQ
WB2JKF
WB2KIW
K2KJI
WA2KKZ
WB2KLT
WB2KMY
WA2KOK
VE2KV
VE2KW
W2KXW
W2LQJ
WA2LQQ
WB2LTJ
K2LWR
WA2LYG
K2LZF
WA2MCB
WA2MJX
WA2MMX
K2MP
WB2MRX
WA2MWI
W2NKP
WB2NCF
WA2NNY
WA2NRJ
K2NV
W2OEZ
WB2OMG
W2OQV
W2ORA
K2OS
WA2OUS
K2OVS
W2OW
W2OWF
K2OWR
K2OY
W2PAU
WA2PBB
W2PGC
WA2PJZ
W2PPS
WA2PVV
WB2QBT
W2QHM
WA2QMH
WB2QOQ
WB2RDE
K2RKW
W2RLV
WB2RRK
WA2RUX
WB2RYY
VE2SH
WA2SLY
WA2SNA
W2SNI
K2SQS
W2SZ
WB2SZK
WA2SZY
WB2TBB
WB2TCQ
WB2TFH
WA2THS
WA2TIF
W2TIN
K2TTI
W2UAD
WA2UBO
W2UBR
WB2UEE
K2ULR
WA2USX
WB2VEG
W2VJN
K2VNV
W2VO
K2VPQ
W2VYQ
W2WGL
WB2WIH
WB2WIK
WB2WLP
WA2WNI
WA2WUL
WA2WVL
WB2WVW
WA2WZU
WB2YBC
K2YCO
WA2YDG
K2YEH
K2YGF
WB2YMW
K2YR
WA2YSM
WA2YTM
W2YX
WB2ZDD
WA2ZJF
K2ZT
K2ZTL
WA2ZWH

-3-

K3ACR
VE3AEA
N3AHI
KA3AJH
VE3AQG
K3ARN
N3AUA
K3AW
WB3AZH
VE3BBO
VE3BDO
W3BID
VE3BQN
WA3BSZ
VE3BYO
VE3CLB
W3CLQ
WA3CPH
W3CSA
WB3CZG
VE3DKW
VE3DTQ
VE3EEH
WB3EFI
WB3ERE
VE3ESE
K3EUG
W3EYF
WA3FFC
VE3FGU
VE3FN
WB3FWF
WB3FXJ
KB3GB
W3GE
VE3HAB
WB3HAL
K3HAM
K3HCE
K3HZO
W3HZU
K3ICH
WB3IHY
W3IIT
W3ILG
K3IVO
W3IWI
K3IWK
W3IY
WB3JUK
WB3JYO
VE3KFD
W3KO
WA3KPP
W3KQR
K3KTY
W3KWH
K3LNZ
W3LP
K3MBM
W3MPK
N3MW
K3MWV
K3NHX
W3ORU
W3PHZ
VE3RL
W3SMK
WA3SXH
K3SZY
AE3T
K3UAV
VE3UH
WA3USC
WA3VCS
W3VUH
W3VXJ
W3WIP
WA3YAN
W3ZR
W3ZZ

prevent those times from showing up in your final logs as extra operating time, enter "0" in response to each time query when you begin or end a post-contest editing session.

25. Finally, run the DUPEPRINT program. That program will print dupesheets from the data files generated by either the LOGGER or DUPECHECKER. In either case, it will sort all call signs either alphabetically or by call area, and print out neat, well-aligned, and fully paginated dupesheets. As we said in the instructions for the DUPECHECKER, the sorting of calls by the DUPEPRINT program takes time, so you will probably want to start it running and then do something else while your computer does all the work that you used to do: duping your logs, and tabulating and calculating the results.

26. Now you're done. You can sit back and relax while your friends and competitors grumble about the long hours they're putting in dupechecking their logs after the big contest.

LOGGING IN SPECIFIC CONTESTS

The LOGGER program is tailored to the general procedures in typical contests. But some contests—especially DX contests—pose special problems. For example, in the ARRL DX Contest (in which U.S. and Canadian stations work only foreign stations), no part of the exchange that W/VE stations receive is a multiplier. W/VE stations determine the country of each station they work from the call sign.

Given the frequent changes in call signs—and the wide variety of call signs used to designate the same country—we suggest that ARRL DX contest operators in the United States and Canada simply enter a standard prefix for each country as a separate item after the numeric exchange (e.g., "59500 JA" for any Japanese station that sends "59500," "57125 DJ" for any West German who sends "57125," etc.). That takes very little extra time, and it assures that the LOGGER will correctly record and score your multipliers. Then the on-line multiplier status reports will be correct, too.

For foreign stations operating the ARRL DX contest, there is no problem with multipliers: the information foreign stations receive from W/VE stations includes the multiplier. Thus, the LOGGER works correctly for foreign stations as it stands.

Either way, as long as the LOGGER is instructed to treat the last half of the exchange as the multiplier, everything works correctly. Enter the correct point value (3 points per contact regardless of band) when you initialize the program before the contest, and the LOGGER will give you an accurate running score (or total score), complete with a full list of the multipliers you've worked on each band in the ARRL DX Contest.

However, the CQ Worldwide DX Contest has further complicating factors: a double multiplier system (zones and countries) PLUS a sliding point scale that depends on where each station that you work is located. We decided against including the code to keep track of all that in the GENERAL PURPOSE CONTEST LOGGER.

Another contest that poses a special problem is the VHF Sweepstakes, in which the scoring is based on the multiplier total *plus 10.* The LOGGER works fine in VHF SS, but you'll have to add 10 to your multiplier and then recompute the final score, correcting the score shown on your summary sheet.

There are a number of contests that require the exchange of consecutive serial numbers. The LOGGER displays a consecutive "record number" for each contact, and that number can be sent as a serial number. In that way, the LOGGER serves the specific requirements of many additional contests.

In general, the LOGGER works for any contest in which there is a multiplier based on the exchange (or just the last part of the exchange), with each QSO worth a specified number of points (which can vary from band to band), and with the final score determined on a QSO-points-times-multiplier basis. The LOGGER will work whether multipliers count just once or there are separate band-multipliers. And its dupechecking feature can be set up for either a once-per-contest or a once-per-band type of dupecheck.

That's pretty versatile, but it doesn't work in all contests. In two cases we decided a particular contest was so popular and specialized in its exchange or scoring procedure that a specialty logging program was justified: Sweepstakes and Field Day.

The Field Day Logger—Operating Notes

Most of the operating procedures for the GENERAL PURPOSE LOGGER apply equally to the FIELD DAY LOGGER and the SWEEPSTAKES LOGGER programs, but there are a few differences that should be noted.

In the Field Day rules, there's no multiplier based on geography. Thus, the FD LOGGER ignores the geographical exchange in computing the score. In Field Day, a contact across town is just as good as one with Ulan Bator, and 100 contacts across town are just as good as contacts with 100 different DXCC countries.

The FD LOGGER asks several questions at the beginning to determine your power multiplier, whether you are a Class D station (i.e., a home station not running emergency power), and the like. If you're a Class D station, the FD LOGGER doesn't count contacts with other stations that send Class D as their category, thus complying with the Field Day rule in that regard.

The FD LOGGER is tailored to the Field Day exchange: entry class (or signal report for foreign stations) and section (or country). You can enter it all on the same line as the call sign, or enter the call sign first for dupechecking purposes.

Another unique feature: when you specify bands, you must say either "PH" or "CW" along with the band, since voice and CW count as separate bands in Field Day. For dupecheck purposes, each is treated separately.

Almost all of the menu commands we described earlier work with the FD LOGGER. You can edit any entry, call up your SCORE or STAT (status

on a single band), delete the last entry with "X" or abort the current one with "–," display the menu with "?," and use "QSY," "SAVE," and "QRT" in the usual way.

The QRX and RATE features were omitted from the FD LOGGER due to that contest's rules. During Field Day, each station may operate for 24 hours from its starting time (or for the full 27 hours of Field Day if certain other requirements are met). Thus, it doesn't matter how many breaks you take during your operating period—you still have to stop at a specified time. You should write down your official starting time so you can comply with the rules by shutting down 24 (or 27) hours later.

Given the immense popularity of Field Day (every year, some 25,000 radio amateurs participate), and because Field Day is usually a group project, we suspect that the FIELD DAY LOGGER will be a widely used program. Some people may hesitate to take their computers off into the countryside, and only the most cavalier would run a computer on a flaky gasoline generator without a good regulator in between, but nevertheless, we think a small, portable computer will soon be as standard a piece of Field Day gear as the generator, the beer, and the insect repellant!

FIELD DAY LOGGER (TRS-80 and CP/M)

```
10 REM FIELD DAY LOGGER (TRS-80 VERSION)
20 REM BY KC6A AND N6NB
30 REM (C) 1983 BY HAYDEN BOOK CO.,INC.
35 CLEAR 1000
40 BL$=CHR$(7):REM NEEDED; DON'T DELETE
52 HI$="":LO$="":REM NEEDED FOR OTHER COMPUTERS
60 CLS:REM CLEAR SCREEN
70 FA$="DUPELOGA":FB$="DUPELOGB":FC$="CONTSTLG"
80 ZA=0:AA=ASC("A")-1:A0=ASC("A")-ASC("0")+26
90 T1$="BAND      QSO'S X Q-PTS = POINTS"
100 T2$="          -----          ------"
110 GOSUB 2000
120 PRINT:A$="":INPUT "NEW CONTEST (N) OR RESUME (R) ";A$
122 CLS
123 INPUT "WANT TO USE THE REAL TIME CLOCK TO LOG TIME";RC$
124 IF RC$="Y" THEN PRINT "BE SURE CLOCK IS SET CORRECTLY."
130 IF A$="N" THEN  190
140 IF A$<>"R" THEN PRINT BL$:GOTO 110
150 GOSUB 2230:REM READ IN FILE
160 IF HC$="Y" THEN GOSUB 1090
170 SS=26*NL+1:GOTO 440
180 REM NEW CONTEST
190 PRINT:INPUT "WANT HARD COPY WHILE LOGGING (Y/N) ";HC$
200 AA$="":IF HC$="Y" THEN INPUT "TURN ON PRINTER. <CR> WHEN READY ";AA$:GOSUB 1
    090
210 Y$="FIELD DAY"
220 PRINT:INPUT "HOW MANY SEPARATE LOG SHEETS DO YOU NEED ";NL:PRINT
230 IF NL<1 THEN PRINT BL$:GOTO 220
240 DIM LN$(NL),QP%(NL),TC%(NL)
250 M1=INT((FRE(0)-2000-800*NL)/6)
260 IF NL*26>=M1 THEN PRINT "TOO MANY";BL$:ERASE LN$,QP%,TC%:GOTO 220
270 DIM C1%(M1),C2%(M1),P%(M1)
275 KK$=""
280 FOR J=1 TO NL
290 PRINT:PRINT "WHAT BAND IS LOG SHEET #";J
300 PRINT "ENTER THE BAND FOLLOWED BY 'PH' FOR PHONE OR 'CW' FOR CW"
```

```
310 INPUT "  (E.G. 20PH, 20CW, 40PH, OR 15CW) ";LN$(J)
312 IF J=1 THEN 320
314 FOR K=1 TO J-1:IF LN$(J)=LN$(K) THEN PRINT "DUPLICATE BAND";BL$:GOTO 290
316 NEXT K
320 IF LEN(LN$(J))>8 THEN PRINT BL$;"NAME TOO LONG":GOTO 290
330 W$=RIGHT$(LN$(J),1):IF W$="H" THEN QP%(J)=1:GOTO 360
340 IF W$="W" THEN QP%(J)=2:GOTO 360
350 PRINT BL$;"INVALID NAME. TRY AGAIN":GOTO 290
360 KK$=KK$+LN$(J)+"  "
362 PRINT:PRINT "BANDS SPECIFIED: ";KK$:PRINT
364 TC%(J)=0:NEXT J
370 T=1:L%=26*NL:SS=L%+1
380 PRINT:PRINT "SPACE IS AVAILABLE FOR APPROXIMATELY";M1;"CALL SIGNS. GOOD LUCK
    !"
390 PRINT:PRINT "IS THIS A CLASS D STATION "
392 INPUT "AT HOME USING NORMAL A.C. POWER (Y/N)";CD$
400 IF CD$<>"Y" THEN IF CD$<>"N" THEN PRINT BL$:GOTO 390
410 PRINT: PRINT "WHAT IS YOUR POWER MULTIPLIER ('1' IF OVER 200 W.; '2' IF BELO
    W"
412 INPUT "200 W., AND '5' IF ALWAYS BELOW 10 W. AND USING BATTERIES) ";PM
420 IF PM<>1 THEN IF PM<>2 THEN IF PM<>5 THEN PRINT BL$:GOTO 410
430 FOR J=1 TO L%:C1%(J)=-32000:P%(J)=0:NEXT J
440 GOSUB 3490
450 LG=1:LH%=0:LP=1:LL=0:LC$=""
460 GOSUB 1150:IF A$="N" THEN Z$=DTE$:REM DATE
470 IF HC$="Y" THEN GOSUB 1110
480 GOSUB 1510:REM BAND
490 MD$=""
500 GOSUB 1290:REM MENU
510 PRINT:PRINT "LOGGER READY FOR QSO #";T;"-";LN$(LG)
520 PRINT "COMMAND OR CALL:";
530 X1%=0:GOSUB 2060:IF X1%=1 THEN 510
540 IF G$="?" THEN GOSUB 1290:GOTO 510
550 IF G$="QSY" THEN GOSUB 1510:GOSUB 1610:GOTO 510
560 IF G$="DATE" THEN GOSUB 1150:GOTO 510
570 IF G$="QRT" THEN GOSUB 3380:GOSUB 1660:GOTO 1790
580 IF G$="SAVE" THEN GOSUB 1670:CLOSE:GOSUB 3490:GOTO 510
590 IF G$="STAT" THEN GOSUB 1610:GOTO 510
600 IF G$="SCORE" THEN GOSUB 3380:GOTO 510
610 IF G$="EDIT" THEN GOTO 2380
620 IF G$="X" THEN GOSUB 1200:GOTO 510
630 GOTO 740
640 PRINT BL$;HI$
650 PRINT STRING$(38,140)
660 PRINT "**  IT'S A DUPE  **  BOO  **  HISS  **"
670 PRINT STRING$(38,131):PRINT LO$
680 GOTO 510
690 PRINT "INVALID ENTRY. PLEASE TRY AGAIN.";BL$:GOTO 510
700 PRINT "ABORTED":RETURN
710 PRINT "CAN'T DELETE";BL$:RETURN
720 PRINT "INVALID QSO #";BL$:GOTO 510
730 REM PROCESS A CALL SIGN
740 CS$=G$:CT$=G$
750 GOSUB 3110:ON X1% GOTO 640,900,690,760,760:REM DECODE CALL & JUMP ON RESULT
760 GOSUB 2090:IF LEN(G$)<>0 THEN 790
770 PRINT " -OK-  CLASS (<CR>  TO ABORT)? ";:GOSUB 2060:IF X1%=1 THEN 510
780 IF G$="" THEN 510
790 CL$=G$:GOSUB 3520:IF X1%=1 THEN 510
800 GOSUB 2090
810 SK$=G$:IF X1%=1 THEN 510
820 IF SK$="" THEN PRINT "SECTION? ";:GOSUB 2060:GOTO 810
830 IF RC$="Y" THEN GOSUB 1182:GOTO 840
832 GOSUB 2090:UTC$=G$:IF UTC$="" THEN PRINT "UTC? ";:GOSUB 2060:UTC$=G$:IF X1%=
    1 THEN 510
840 RN=T:GOSUB 980:REM WRITE LOG TO DISK
850 IF HC$="Y" THEN GOSUB 1040:REM WRITE TO PRINTER
860 LH%=H%:GOSUB 1070:REM ENTER IN DUPE SHEET
```

```
870 IF CD$="Y" THEN IF LC$="D" THEN 890
880 TC%(LG)=TC%(LG)+1
890 T=T+1:GOTO 510
900 PRINT:PRINT TAB(2)CT$;" NEEDED ON:"
910 FOR J=1 TO NL
920 H%=E(4)+J*26-26:GOSUB 3320:REM DUPE SEARCH
930 IF X1%=1 THEN 960
940 IF J=LG THEN PRINT TAB(10)"* ";ELSE PRINT "           ";
950 PRINT LN$(J)
960 NEXT J:PRINT:GOTO 510
970 REM WRITE LOG TO DISK
980 PRINT "...LOGGING"
990 LSET D1$=DTE$:LSET TM$=UTC$:LSET KL$=CT$
1000 LSET B1$=LN$(LG):LSET M1$=MD$:LSET E1$=SK$:LSET C1$=CL$
1010 PUT 3,RN
1020 RETURN
1030 REM WRITE LOG TO PRINTER
1040 LPRINT T;TAB(6)DTE$;TAB(17)UTC$;TAB(24)LN$(LG);TAB(30)MD$;TAB(36)CT$;TAB(48
     )C1$;SK$
1050 RETURN
1060 REM ENTER IN DUPE LIST
1070 IF X1%=5 THEN L%=L%+1:P%(H%)=L%:H%=L%
1080 C1%(H%)=E1%:C2%(H%)=E2%:P%(H%)=0:RETURN
1090 REM PRINTER INIT, IF USED
1100 RETURN
1110 LPRINT:LPRINT:LPRINT TAB(5)"CONTEST: ";Y$
1120 LPRINT TAB(5)"DATE:    ";Z$
1130 LPRINT:RETURN
1140 REM SET DATE
1150 IF RC$="Y" THEN 1182
1152 GOSUB 2090
1160 IF G$="" THEN PRINT:G$="":INPUT "WHAT IS THE DATE (MM/DD/YY) ";G$
1170 IF LEN(G$)>8 THEN PRINT "DATE TOO LONG";BL$:G$="":GOTO 1160
1180 DTE$=G$:RETURN
1182 DTE$=LEFT$(TIME$,8):UTC$=MID$(TIME$,10,2)+MID$(TIME$,13,2)
1184 PRINT "UTC=";UTC$:RETURN
1190 REM X DELETES LAST ENTRY
1200 IF LH%=0 THEN 710
1210 IF LH%<SS THEN IF P%(LH%)=0 THEN C1%(LH%)=-32000:GOTO 1230
1220 P%(LH%)=0:L%=L%-1
1230 T=T-1:LH%=0
1240 IF CD$="Y" THEN IF LC$="D" THEN 1260
1250 TC%(LG)=TC%(LG)-1
1260 IF HC$="Y" THEN LPRINT "-- PREVIOUS CALL DELETED --"
1270 PRINT " DELETED":RETURN
1280 REM MENU
1290 CLS
1300 PRINT HI$
1305 PRINT "            FIELD DAY LOGGER             "
1310 PRINT TAB(11)STRING$(16,131)
1320 PRINT " '?' DISPLAYS MENU                       "
1330 PRINT " 'QSY <BAND>' CHANGES BAND               "
1340 PRINT " 'DATE <MM/DD/YY>' CHANGES DATE          "
1350 PRINT " 'QRT' SAVES ALL CALLS AND EXITS         "
1360 PRINT " 'SAVE' SAVES ALL CALLS AND CONTINUES    "
1370 PRINT "          (NOTE: ALLOW 2-3 MIN FOR SAVE) "
1380 PRINT " 'STAT' GIVES STATUS ON CURRENT BAND     "
1390 PRINT " 'SCORE' GIVES RUNNING TOTAL SCORE       "
1400 PRINT " 'EDIT <QSO #>' EDITS ANY LOGGED ENTRY   "
1410 PRINT " 'X' DELETES THE LAST ENTRY              "
1420 PRINT " '*' ABORTS AN ENTRY                     "
1425 CR$="":INPUT "<ENTER> TO CONTINUE";CR$
1427 PRINT "            MENU (CONTINUED)"
1428 PRINT TAB(10)STRING$(16,131)
1430 PRINT " ENTER CALL ONLY FOR DUPE CHECK          "
1440 PRINT " ENTER CALL AND '?' TO SEE BANDS ON      "
1444 PRINT "          WHICH A STATION IS NEEDED      "
1450 PRINT " ENTRY FORMAT: 'CALL CLASS SECT <UTC>' "
```

```
1460 PRINT "          (E.G.  W2RQ 13A NNJ 1345)      "
1470 PRINT "          (OR:   JA1MIN 579 JAPAN 1345) "
1480 PRINT " <> DENOTES OPTIONAL ENTRIES             ";LO$
1490 RETURN
1500 REM QSY
1510 GOSUB 2090
1520 IF NL=1 THEN 1590
1530 IF G$="" THEN PRINT:INPUT "WHAT BAND (LOG SHEET) DO YOU WANT ";G$
1540 FOR J=1 TO NL: IF G$=LN$(J) THEN 1580
1550 NEXT J
1560 PRINT "NO SUCH LOG. VALID LOGS ARE:";BL$
1570 FOR J=1 TO NL:PRINT TAB(5)LN$(J):NEXT J:G$="":GOTO 1530
1580 LG=J:IF HC$="Y" THEN GOSUB 2180
1590 RETURN
1600 REM STATUS
1610 CLS:PRINT:PRINT "DATE= ";DTE$;
1620 PRINT "  BAND= ";LN$(LG)
1630 PRINT "  CONTACTS= ";TC%(LG)
1640 PRINT:RETURN
1650 REM QRT
1660 CLOSE 3
1670 OPEN "O",1,FA$
1680 OPEN "R",2,FB$,6
1690 FIELD 2,2 AS C1$,2 AS C2$,2 AS P$
1700 PRINT#1,HC$:PRINT#1,T:PRINT#1,Y$:PRINT#1,Z$:PRINT#1,NL:PRINT#1,L%:PRINT#1,M
     1
1710 FOR J=1 TO NL:PRINT#1,LN$(J):NEXT J
1715 PRINT#1,ZA
1720 FOR J=1 TO L%
1730 LSET C1$=MKI$(C1%(J)):LSET C2$=MKI$(C2%(J)):LSET P$=MKI$(P%(J))
1740 PUT 2,J:NEXT J:CLOSE 2
1750 FOR J=1 TO NL:PRINT#1,QP%(J);",";TC%(J):NEXT J
1760 PRINT#1,CD$:PRINT#1,PM
1770 RETURN
1780 REM WRITE SCORE TO LOG#1
1790 PRINT #1,BL$:REM NEW PAGE
1800 PRINT#1,TAB(14)"SCORE SUMMARY":PRINT#1,:PRINT#1,T1$
1810 IF HC$="Y" THEN LPRINT:LPRINT TAB(14)"SCORE SUMMARY":LPRINT:LPRINT T1$
1820 X2=0:X3=0:X4=0
1830 FOR J=1 TO NL
1840 X4=X4+TC%(J):X1=TC%(J)*QP%(J):X2=X2+X1
1850 PRINT#1,LN$(J);TAB(11)TC%(J);TAB(20)QP%(J);TAB(27)X1
1860 IF HC$="Y" THEN LPRINT LN$(J);TAB(11)TC%(J);TAB(20)QP%(J);TAB(27)X1
1870 NEXT J
1880 PRINT#1,T2$:PRINT#1,"TOTAL";TAB(11)X4;TAB(27)X2
1890 PRINT#1,"POWER MULTIPLIER";TAB(25)"X ";PM:PRINT#1,TAB(27)"------"
1900 PRINT#1,"TOTAL POINTS";TAB(27)PM*X2:PRINT#1,
1910 IF HC$="N" THEN 1950
1920 LPRINT T2$:LPRINT "TOTAL";TAB(11)X4;TAB(27)X2:LPRINT
1930 LPRINT "POWER MULTIPLIER";TAB(25)"X ";PM:LPRINT TAB(27)"------"
1940 LPRINT "TOTAL POINTS =";TAB(27)PM*X2
1950 PRINT#1,:PRINT#1,"THIS IS A TOTAL OF THE QSO POINTS EARNED IN THIS LOG ONLY
     ."
1960 PRINT#1,"FOR THE FINAL SCORE, ADD THE QSO POINTS FROM ANY OTHER"
1970 PRINT#1,"LOGS NOT INCLUDED HERE PLUS ANY BONUS POINTS EARNED."
1980 CLOSE
1990 END
2000 CLS:PRINT CHR$(23):PRINT:REM PRINT TITLE
2010 PRINT:PRINT
2020 PRINT TAB(3)"FIELD DAY LOGGING PROGRAM"
2030 PRINT "(C) 1983 BY HAYDEN BOOK CO.,INC."
2040 PRINT:PRINT:RETURN
2050 REM INPUT SCAN ROUTINE
2060 LINE INPUT GS$:LL=LEN(GS$)
2070 IF INSTR(GS$,"*")>0 THEN GOSUB 700:X1%=1:RETURN
2080 LP=1
2090 IF LP>LL THEN 2120:REM 2ND ENTRY
```

```
2100 FOR LT=LP TO LL:IF MID$(GS$,LT,1)<>" " THEN 2130
2110 NEXT LT
2120 G$="":RETURN
2130 LP=LT
2140 FOR LT=LP TO LL:IF MID$(GS$,LT,1)=" " THEN 2170
2150 NEXT LT
2160 LT=LT+1
2170 G$=MID$(GS$,LP,LT-LP):LP=LT:RETURN
2180 LPRINT
2190 LPRINT TAB(5)"QSY TO: ";LN$(LG)
2200 RETURN
2210 LL=LEN(GS$):GOTO 2080
2220 REM RESUME
2230 OPEN "I",1,FA$
2240 OPEN "R",2,FB$,6
2250 FIELD 2,2 AS C1$,2 AS C2$,2 AS P$
2260 INPUT#1,HC$:INPUT#1,T:INPUT#1,Y$:INPUT#1,Z$:INPUT#1,NL:INPUT#1,L%:INPUT#1,M
     1
2270 DIM LN$(NL),QP%(NL),TC%(NL)
2280 DIM C1%(M1),C2%(M1),P%(M1)
2290 FOR J=1 TO NL:INPUT#1,LN$(J):NEXT J
2295 INPUT#1,ZA
2300 FOR J=1 TO L%
2310 GET 2,J
2320 C1%(J)=CVI(C1$):C2%(J)=CVI(C2$):P%(J)=CVI(P$)
2330 NEXT J
2340 FOR J=1 TO NL:INPUT#1,QP%(J),TC%(J):NEXT J
2350 INPUT#1,CD$:INPUT#1,PM
2360 CLOSE:RETURN
2370 REM EDIT A RECORD
2380 GOSUB 2090:IF G$="" THEN INPUT "WHAT QSO# DO YOU WANT TO EDIT ";G$
2390 X2=INT(VAL(G$))
2400 IF X2>0 THEN IF X2<T THEN 2420
2410 GOTO 720
2420 CLS
2430 GET 3,X2:Y$(2)=D1$:Y$(3)=TM$:Y$(4)=KL$:Y$(5)=B1$:Y$(6)=M1$:Y$(7)=C1$:Y$(8)=
     E1$
2440 GOSUB 2770
2450 PRINT:A$="":INPUT "CHANGE THIS RECORD (Y/N) ";A$
2460 IF A$<>"Y" THEN CLS:GOTO 510
2470 GS$=B1$:GOSUB 2210
2480 FOR J=1 TO NL:IF G$=LN$(J) THEN 2500
2490 NEXT J
2500 LX=LG:LY=J:LK=LY
2510 GS$=C1$:GOSUB 2210:CL$=G$:GOSUB 3520:LB$=LC$
2520 PRINT:PRINT TAB(12)"INSTRUCTIONS":PRINT
2530 PRINT "HIT <CR> TO LEAVE AN ITEM UNCHANGED."
2540 PRINT "TO CHANGE, ENTER NEW VALUE <CR>."
2550 PRINT "TO DELETE THIS CALL, CHANGE CALL TO '*'."
2560 PRINT:PRINT TAB(5)"DATE:";TAB(14)Y$(2);:A$="":INPUT A$:IF A$<>"" THEN Y$(2)
     =A$
2570 PRINT TAB(5)"TIME:";TAB(14)Y$(3);:A$="":INPUT A$:IF A$<>""THEN Y$(3)=A$
2580 PRINT TAB(5)"CALL:";TAB(14)Y$(4);:A$="":INPUT A$:IF A$<>""THEN Y$(4)=A$
2590 PRINT TAB(5)"BAND:";TAB(14)Y$(5);:A$="":INPUT A$:IF A$<>"" THEN Y$(5)=A$
2600 PRINT TAB(5)"CLASS:";TAB(14)Y$(7);:A$="":INPUT A$:IF A$<>"" THEN Y$(7)=A$
2610 PRINT TAB(5)"SECTION:";TAB(14)Y$(8);:A$="":INPUT A$:IF A$<>"" THEN Y$(8)=A$
2620 PRINT:A$="":INPUT "OK (Y/N) ";A$:X1%=4
2630 IF A$<>"Y" THEN CLS:GOSUB 700:GOTO 510
2640 GS$=Y$(7):GOSUB 2210:CL$=G$:GOSUB 3520
2650 IF Y$(5)<>B1$ THEN GOSUB 2930:GOTO 2690
2660 LG=LK
2670 IF Y$(4)<>KL$ THEN GOSUB 2840:GOSUB 2870:GOSUB 3000:GOSUB 3040:GOTO 2690
2680 IF Y$(7)<>C1$ THEN GOSUB 3000:GOSUB 3040
2690 LG=LX
2700 ON X1% GOTO 640,3080,3080,2710,2710,3090
2710 LSET D1$=Y$(2):LSET TM$=Y$(3):LSET KL$=Y$(4)
2720 LSET B1$=Y$(5):LSET M1$=Y$(6):LSET C1$=Y$(7):LSET E1$=Y$(8)
```

```
2730 PUT 3,X2
2740 IF HC$<>"Y" THEN 2760
2750 LPRINT "  ** EDIT TO REC #";X2;": ";D1$;" ";TM$;" ";KL$;" ";B1$;" ";C1$;E1$
2760 GOTO 510
2770 PRINT:PRINT  TAB(5)"DATE:";TAB(14)Y$(2)
2780 PRINT TAB(5)"TIME:";TAB(14)Y$(3)
2790 PRINT TAB(5)"CALL:";TAB(14)Y$(4)
2800 PRINT TAB(5)"BAND:";TAB(14)Y$(5)
2810 PRINT TAB(5)"EXCHANGE:";TAB(14)Y$(7);Y$(8)
2820 RETURN
2830 REM CALL CHANGED
2840 IF Y$(4)="*" THEN RETURN
2850 GS$=Y$(4):GOSUB 2210:CS$=G$:GOSUB 3110:IF X1%<4 THEN RETURN:REM CHECK IF NE
     W CALL IS DUPE
2860 GOSUB 1070:RETURN
2870 IF X1%<4 THEN RETURN
2880 IF LEFT$(KL$,1)="*" THEN RETURN
2890 GS$=KL$:GOSUB 2210:CS$=G$
2900 GOSUB 3110:C1%(H%)=-32001:C2%(H%)=-32000:REM DELETE OLD CALL
2910 X1%=4:RETURN
2920 REM CHANGE BAND
2930 FOR LK=1 TO NL:IF Y$(5)=LN$(LK) THEN 2960
2940 NEXT LK
2950 X1%=6:RETURN
2960 LG=LK:GOSUB 2840:IF X1%<4 THEN RETURN
2970 LG=LY:GOSUB 2870
2980 GOSUB 3000:LG=LK:GOSUB 3040:RETURN
2990 REM CHANGE EXCHANGE
3000 IF LEFT$(KL$,1)="*" THEN RETURN
3010 IF CD$="Y" THEN IF LB$="D" THEN 3030
3020 TC%(LG)=TC%(LG)-1
3030 RETURN
3040 IF Y$(4)="*" THEN RETURN
3050 IF CD$="Y" THEN IF LC$="D" THEN 3070
3060 TC%(LG)=TC%(LG)+1
3070 RETURN
3080 PRINT BL$;"INVALID CALL SIGN":GOTO 510
3090 PRINT BL$;"INVALID BAND":GOTO 510
3100 REM DUPE CHECKER: X1%=1=DUPE: 2=QUERY: 3=INVALID CALL: 4,5=OK
3110 X1%=4:E(5)=0:E(6)=0:K=1:IF LEN(CS$)>10 THEN X1%=3:RETURN
3120 IF RIGHT$(CS$,1)="?" THEN CS$=LEFT$(CS$,LEN(CS$)-1):X1%=2:CT$=CS$
3130 IF LEN(CS$)<4 THEN X1%=3:RETURN
3140 IF MID$(CS$,3,1)<"A" THEN 3170
3150 IF MID$(CS$,2,1)=>"A" THEN X1%=3:RETURN
3160 CS$=" "+CS$:K=2:E(1)=0
3170 FOR J=K TO LEN(CS$)
3180 W$=MID$(CS$,J,1):IF W$="/" THEN 3230
3190 E(J)=ASC(W$)-AA:IF E(J)>26 THEN X1%=3:RETURN
3200 IF W$<"0" THEN X1%=3:RETURN
3210 IF E(J)<1 THEN E(J)=E(J)+A0:IF E(J)>36 THEN X1%=3:RETURN
3220 NEXT J
3230 IF J>7 THEN X1%=3:RETURN
3240 IF J-K<4 THEN X1%=3:RETURN
3250 FOR J=4 TO 6:IF E(J)>26 THEN X1%=3:RETURN
3260 NEXT J
3270 E1%=-32000+E(1)*1369+E(2)*37+E(3)
3280 E2%=-32000+E(4)*1369+E(5)*37+E(6)
3290 IF X1%=2 THEN RETURN
3300 H%=E(4)+LG*26-26
3310 REM SEARCH DUPE STRING
3320 IF C1%(H%)=-32000 THEN X1%=4:RETURN
3330 IF C2%(H%)<>E2% THEN 3350
3340 IF C1%(H%)=E1% THEN X1%=1:RETURN
3350 IF P%(H%)=0 THEN X1%=5:RETURN
3360 H%=P%(H%):GOTO 3330
3370 REM SCORE
3380 CLS:PRINT:PRINT TAB(22)"SCORE SUMMARY":PRINT
3390 PRINT T1$
```

```
3400 X2=0:X3=0:X4=0
3410 FOR J=1 TO NL
3420 X4=X4+TC%(J):X1=TC%(J)*QP%(J):X2=X2+X1
3430 PRINT LN$(J);TAB(11)TC%(J);TAB(20)QP%(J);TAB(27)X1
3440 NEXT J
3450 PRINT T2$
3460 PRINT "TOTAL";TAB(11)X4;TAB(27)X2
3470 PRINT:PRINT TAB(10)"SCORE = POWER MULT (";PM;") X ";X2;" = ";X2*PM
3480 PRINT:RETURN
3490 OPEN "R",3,FC$,47
3500 FIELD 3,8 AS D1$,4 AS TM$,10 AS KL$,5 AS B1$,5 AS M1$,5 AS C1$,10 AS E1$
3510 RETURN
3520 IF LEN(CL$)=>2 THEN IF LEN(CL$)<=3 THEN 3540
3530 PRINT BL$;"INVALID CLASS (OR REPORT). RE-ENTER? ";:GOSUB 2060:CL$=G$:IF X1%
     =1 THEN RETURN ELSE 3520
3540 LC$=LEFT$(CL$,1):IF LC$<="9" THEN IF LC$>="0" THEN 3560
3550 GOTO 3530
3560 LC$=RIGHT$(CL$,1):IF LC$>="A" THEN IF LC$>"E" THEN 3530
3570 LH%=0:RETURN
```

FIELD DAY LOGGER (Apple)

```
10  REM  FIELD DAY LOGGER (APPLE)
20  REM  COPYRIGHT 1983 BY HAYDEN BOOK CO.,INC.
40 BL$ =  CHR$ (7): REM  BELL
50 D$ =  CHR$ (4):Q$ =  CHR$ (34)
52 LP$ = D$ + "PR#1":SCN$ = D$ + "PR#0"
54 RD$ = D$ + "READ ":WR$ = D$ + "WRITE "
56 CL$ = D$ + "CLOSE ":OP$ = D$ + "OPEN "
57 DL$ = D$ + "DELETE "
58 DV$ = ",D1"
60  HOME
70 F1$ = "DUPELOGA"
72 F2$ = "DUPELOGB":R2$ = ",L19"
74 F3$ = "CONTSTLG":R3$ = ",L58"
80 AA =  ASC ("A") - 1:A0 =  ASC ("A") -  ASC ("0") + 26:ZA = 0
90 T1$ = "BAND      QSO'S X Q-PTS = POINTS"
100 T2$ = "          -----          ------"
105  DIM Y$(20)
110  GOSUB 2000
115  PRINT : PRINT "ARE YOU USING A REAL TIME CLOCK FOR"
116  INPUT "LOGGING THE TIME (Y/N)? ";RC$
120  PRINT : INPUT "NEW CONTEST (N) OR RESUME (R)? ";A$
130  IF A$ = "N" THEN 190
140  IF A$ <  > "R" THEN  PRINT BL$: GOTO 110
150  GOSUB 2230: REM   READ IN FILE
155  PRINT : PRINT "RESUMING ";Y$;" OF ";Z$
160  IF HC$ = "Y" THEN  GOSUB 1090
170 SS = 26 * NL + 1: GOTO 450
180  REM  NEW CONTEST
190  PRINT : INPUT "WANT HARD COPY WHILE LOGGING (Y/N)? ";HC$
200  IF HC$ = "Y" THEN  INPUT "TURN ON PRINTER. <CR> WHEN READY ";AA$: GOSUB 1090
210 Y$ = "ARRL FIELD DAY"
220  PRINT : INPUT "HOW MANY SEPARATE LOG SHEETS DO YOU NEED? ";NL: PRINT
230  IF NL < 1 THEN  PRINT BL$: GOTO 220
240  DIM LN$(NL),QP%(NL),TC%(NL)
242 M1 =  FRE (0): IF M1 < 0 THEN M1 = M1 + 65536
```

```
250 M1 =  INT ((M1 - 2000 - 800 * NL) / 6)
260  IF NL * 26 >  = M1 THEN  PRINT "TOO MANY";BL$: END
270  DIM C1%(M1),C2%(M1),P%(M1)
274 XX$ = ""
280  FOR J = 1 TO NL
290  PRINT : PRINT "WHAT BAND IS LOG SHEET #";J
300  PRINT "ENTER THE BAND FOLLOWED BY 'PH'"
301  PRINT "  FOR PHONE OR 'CW' FOR CW"
310  INPUT "  (E.G. 20PH, 20CW, 40PH, OR 15CW)? ";LN$(J)
314  IF J = 1 THEN 320
315  FOR K = 1 TO J - 1: IF LN$(J) = LN$(K) THEN  PRINT "DUPLICATE BAND";BL$: GOTO
 290
316  NEXT K
320  IF  LEN (LN$(J)) > 8 THEN  PRINT BL$;"NAME TOO LONG": GOTO 290
325  IF  LEN (LN$(J)) < 3 THEN 350
330 W$ =  RIGHT$ (LN$(J),2): IF W$ = "PH" THEN QP%(J) = 1: GOTO 360
340  IF W$ = "CW" THEN QP%(J) = 2: GOTO 360
350  PRINT BL$;"INVALID NAME. TRY AGAIN": GOTO 290
360 XX$ = XX$ + " " + LN$(J)
362  PRINT : PRINT "BANDS SPECIFIED SO FAR: ";XX$: PRINT
364 TC%(J) = 0: NEXT J
370 T = 1:L% = 26 * NL:SS = L% + 1
380  PRINT : PRINT "SPACE IS AVAILABLE FOR APPROXIMATELY": PRINT M1;" CONTACTS. GO
OD LUCK!"
390  PRINT : PRINT "IS THIS A CLASS D STATION--"
392  INPUT "AT HOME ON NORMAL A.C. POWER? (Y/N) ";CD$
400  IF CD$ <  > "Y" THEN  IF CD$ <  > "N" THEN  PRINT BL$: GOTO 390
410  PRINT : PRINT "WHAT IS YOUR POWER MULTIPLIER": INPUT "      (1, 2, OR 5)? ";PM
420  IF PM <  > 1 THEN  IF PM <  > 2 THEN  IF PM <  > 5 THEN  PRINT BL$: GOTO 390
430  FOR J = 1 TO L%:C1%(J) =  - 32000:P%(J) = 0: NEXT J
450 LG = 1:LH% = 0:LP = 1:LL = 0:LC$ = ""
460  GOSUB 1150: IF A$ = "N" THEN Z$ = DTE$: REM  DATE
470  IF HC$ = "Y" THEN  GOSUB 1110
480  GOSUB 1510: REM  BAND
490 MD$ = ""
500  GOSUB 1290: REM  MENU
510  PRINT : PRINT "LOGGER READY FOR QSO #";T;".  BAND: ";LN$(LG)
520  PRINT "COMMAND OR CALL";
530 X1% = 0: GOSUB 2060: IF X1% = 1 THEN 510
540  IF G$ = "?" THEN  GOSUB 1290: GOTO 510
550  IF G$ = "QSY" THEN  GOSUB 1510: GOSUB 1610: GOTO 510
560  IF G$ = "DATE" THEN  GOSUB 1150: GOTO 510
570  IF G$ = "QRT" THEN  GOSUB 3380: GOSUB 1660: GOSUB 1790: GOTO 1990
580  IF G$ = "SAVE" THEN  GOSUB 1660: GOSUB 1980: GOTO 510
590  IF G$ = "STAT" THEN  GOSUB 1610: GOTO 510
600  IF G$ = "SCORE" THEN  GOSUB 3380: GOTO 510
610  IF G$ = "EDIT" THEN  GOTO 2380
620  IF G$ = "X" THEN  GOSUB 1200: GOTO 510
630  GOTO 740
640  PRINT BL$: FLASH
650  PRINT "**********************************"
660  PRINT "**  IT'S A DUPE  **  BOO  HISS  **"
670  PRINT "**********************************"
680  NORMAL : GOTO 510
690  PRINT "INVALID ENTRY. PLEASE TRY AGAIN.";BL$: GOTO 510
700  PRINT "ABORTED": RETURN
710  PRINT "CAN'T DELETE";BL$: RETURN
720  PRINT "INVALID QSO #";BL$: GOTO 510
730  REM  PROCESS A CALL SIGN
740 CS$ = G$:CT$ = G$
750  GOSUB 3110: ON X1% GOTO 640,900,690,760,760: REM  DECODE CALL & JUMP ON RESUL
T
760  GOSUB 2090: IF  LEN (G$) <  > 0 THEN 790
770  PRINT " -OK- ";
775  PRINT "CLASS ";: GOSUB 2060: IF X1% = 1 THEN 510
790 CA$ = G$: GOSUB 3520: IF X1% = 1 THEN 510
800  GOSUB 2090
810 SK$ = G$: IF X1% = 1 THEN 510
```

```
 815  IF  LEN (SK$) > 10 THEN SK$ =  LEFT$ (SK$,10)
 820  IF SK$ = "" THEN  PRINT "SECTION ";: GOSUB 2060: GOTO 810
 830  IF RC$ = "Y" THEN  GOSUB 1182: GOTO 840
 832  GOSUB 2090:UTC$ = G$: IF UTC$ = "" THEN  PRINT "UTC ";: GOSUB 2060:UTC$ = G$:
 IF X1% = 1 THEN 510
 840 RN = T: GOSUB 980: REM  WRITE LOG TO DISK
 850  IF HC$ = "Y" THEN  GOSUB 1040: REM  WRITE TO PRINTER
 860 LH% = H%: GOSUB 1070: REM  ENTER IN DUPE SHEET
 870  IF CD$ = "Y" THEN  IF LC$ = "D" THEN 890
 880 TC%(LG) = TC%(LG) + 1
 890 T = T + 1: GOTO 510
 900  PRINT : PRINT  TAB( 2)CT$;" NEEDED ON:"
 910  FOR J = 1 TO NL
 920 H% = E(4) + J * 26 - 26: GOSUB 3320: REM  DUPE SEARCH
 930  IF X1% = 1 THEN 960
 940  PRINT  TAB( 11);: IF J = LG THEN  FLASH
 950  PRINT LN$(J): NORMAL
 960  NEXT J: PRINT : GOTO 510
 970  REM  WRITE LOG TO DISK
 980  PRINT "..LOGGING"
 990  PRINT OP$;F3$;R3$;DV$
 992  PRINT WR$;F3$;",R";RN
1000  GOSUB 3900
1010  PRINT CL$
1020  RETURN
1030  REM  WRITE LOG TO PRINTER
1040  PRINT LP$: PRINT T; SPC( 5 -  LEN ( STR$ (T)));: GOSUB 3900: PRINT SCN$
1050  RETURN
1060  REM  ENTER IN DUPE LIST
1070  IF X1% = 5 THEN L% = L% + 1:P%(H%) = L%:H% = L%
1080 C1%(H%) = E1%:C2%(H%) = E2%:P%(H%) = 0: RETURN
1090  REM  PRINTER INIT, IF USED
1100  RETURN
1110  PRINT LP$: PRINT : PRINT : PRINT  SPC( 4)"CONTEST: ";Y$
1120  PRINT  SPC( 4)"DATE:    ";Z$
1130  PRINT : PRINT SCN$: RETURN
1140  REM  SET DATE
1150  IF RC$ = "Y" THEN  GOTO 1182
1160  IF G$ = "" THEN  PRINT : PRINT "WHAT IS THE DATE (MM/DD/YY) ";: GOSUB 2060:
GOTO 1160
1170  IF  LEN (G$) > 8 THEN  PRINT "DATE TOO LONG";BL$:G$ = "": GOTO 1160
1180 DTE$ = G$: RETURN
1182  PRINT D$;"IN#1": INPUT "DAY/DATE/TIME = ";G$: PRINT D$;"IN#0":DTE$ =  MID$ (
G$,3,8):UTC$ =  MID$ (G$,12,2) +  MID$ (G$,15,2)
1184  RETURN
1190  REM  X DELETES LAST ENTRY
1200  IF LH% = 0 THEN 710
1210  IF LH% < SS THEN  IF P%(LH%) = 0 THEN C1%(LH%) =  - 32000: GOTO 1230
1220 P%(LH%) = 0:L% = L% - 1
1230 T = T - 1:LH% = 0
1240  IF CD$ = "Y" THEN  IF LC$ = "D" THEN 1260
1250 TC%(LG) = TC%(LG) - 1
1260  IF HC$ = "Y" THEN  PRINT LP$: PRINT "-- PREVIOUS CALL DELETED --": PRINT SCN
$
1270  PRINT " DELETED": RETURN
1280  REM  MENU
1290  HOME : INVERSE
1300  PRINT "       ARRL FIELD DAY LOGGER MENU        "
1310  PRINT "                                         "
1320  PRINT " '?' DISPLAYS MENU                       "
1330  PRINT " 'QSY <BAND>' CHANGES BAND               "
1340  PRINT " 'DATE <MM/DD/YY>' CHANGES DATE          "
1350  PRINT " 'QRT' SAVES ALL CALLS AND EXITS         "
1360  PRINT " 'SAVE' SAVES ALL CALLS AND CONTINUES    "
1370  PRINT "         (NOTE: ALLOW 2-3 MIN FOR SAVE)  "
1380  PRINT " 'STAT' GIVES STATUS ON CURRENT BAND     "
1390  PRINT " 'SCORE' GIVES RUNNING TOTAL SCORE       "
1400  PRINT " 'EDIT <QSO #>' EDITS ANY LOGGED ENTRY   "
```

```
1410  PRINT " 'X' DELETES THE LAST ENTRY                "
1420  PRINT " '-' ABORTS AN ENTRY                       "
1450  PRINT " ENTRY FORMAT: 'CALL CLASS SECT <UTC>' "
1460  PRINT "          (E.G.  W2RQ 13A NNJ 1345)        "
1470  PRINT "          (OR:   JA1MIN 579 JAPAN 1345)  "
1472  PRINT " ENTER THE CALL ONLY FOR A DUPE CHECK  "
1474  PRINT " ENTER CALL? FOR BANDS STN NEEDED ON   "
1480  PRINT " <> DENOTES OPTIONAL ENTRIES               "
1490  NORMAL : RETURN
1500  REM  QSY
1510  GOSUB 2090
1520  IF NL = 1 THEN 1590
1530  IF G$ = "" THEN  PRINT : INPUT "WHAT BAND (LOG SHEET) DO YOU WANT? ";G$
1540  FOR J = 1 TO NL: IF G$ = LN$(J) THEN 1580
1550  NEXT J
1560  PRINT "NO SUCH LOG. VALID LOGS ARE:";BL$
1570  FOR J = 1 TO NL: PRINT  TAB( 5)LN$(J): NEXT J:G$ = "": GOTO 1530
1580 LG = J: IF HC$ = "Y" THEN  GOSUB 2180
1590  RETURN
1600  REM  STATUS
1610  HOME : PRINT : PRINT "DATE= ";DTE$;
1620  PRINT "  BAND= ";LN$(LG)
1630  PRINT "  CONTACTS= ";TC%(LG)
1640  PRINT : RETURN
1650  REM  QRT
1660  PRINT OP$;F1$;DV$
1670  PRINT DL$;F1$
1680  PRINT OP$;F1$
1690  PRINT WR$;F1$
1700  PRINT HC$: PRINT T: PRINT Y$: PRINT Z$: PRINT NL: PRINT L%: PRINT M1
1710  FOR J = 1 TO NL: PRINT LN$(J): NEXT J: PRINT ZA
1750  FOR J = 1 TO NL: PRINT QP%(J): PRINT TC%(J): NEXT J
1760  PRINT CD$: PRINT PM
1770  RETURN
1780  REM  WRITE SCORE TO DUPI.LOG
1790  PRINT BL$: REM  NEW PAGE
1800  PRINT Q$; SPC( 13)"SCORE SUMMARY";Q$: PRINT : PRINT Q$;T1$;Q$
1820  GOSUB 3600
1880  PRINT Q$;T2$;Q$
1881  PRINT "TOTAL      ";X4;
1882  PRINT  SPC( 16 -  LEN ( STR$ (X4)))X2
1890  PRINT "POWER MULTIPLIER          X ";PM
1892  PRINT Q$; SPC( 26)"------";Q$
1900  PRINT "TOTAL POINTS"; SPC( 14)PM * X2: PRINT
1950  PRINT : PRINT "THIS IS A TOTAL OF THE QSO POINTS EARNED IN THIS LOG ONLY."
1960  PRINT "FOR THE FINAL SCORE, ADD THE QSO POINTS FROM ANY OTHER"
1970  PRINT "LOGS NOT INCLUDED HERE PLUS ANY BONUS POINTS EARNED."
1975  PRINT "!": REM  EOF MARK
1980  PRINT CL$
1981  PRINT OP$;F2$;DV$
1982  PRINT DL$;F2$
1983  PRINT OP$;F2$;R2$;DV$
1984  FOR J = 1 TO L%
1985  PRINT WR$;F2$;",R";J
1986  PRINT C1%(J): PRINT C2%(J): PRINT P%(J)
1987  NEXT J
1988  PRINT CL$: RETURN
1990  IF HC$ = "Y" THEN  PRINT LP$: GOSUB 3385: PRINT SCN$
1995  END
2000  HOME : PRINT : INVERSE : REM  PRINT TITLE
2020  PRINT "     ARRL FIELD DAY LOGGING PROGRAM     "
2025  PRINT "           BY KC6A AND N6NB             "
2030  PRINT "   (C) 1983 BY HAYDEN BOOK CO., INC.    "
2040  NORMAL : PRINT : PRINT : RETURN
2050  REM  INPUT SCAN ROUTINE
2060  INPUT GS$:LL =  LEN (GS$): IF LL = 0 THEN 2120
2070  FOR J = 1 TO LL: IF  MID$ (GS$,J,1) = "-" THEN X1% = 1: GOTO 700
2071  NEXT J
```

```
2080 LP = 1
2090  IF LP > LL THEN 2120: REM  2ND ENTRY
2100  FOR LT = LP TO LL: IF  MID$ (GS$,LT,1) <  > " " THEN 2130
2110  NEXT LT
2120 G$ = "": RETURN
2130 LP = LT
2140  FOR LT = LP TO LL: IF  MID$ (GS$,LT,1) = " " THEN 2170
2150  NEXT LT
2160 LT = LT + 1
2170 G$ =  MID$ (GS$,LP,LT - LP):LP = LT: RETURN
2180  PRINT LP$: PRINT
2190  PRINT  SPC( 4)"QSY TO: ";LN$(LG)
2200  PRINT SCN$: RETURN
2220  REM  RESUME
2230  PRINT OP$;F1$;DV$
2240  PRINT RD$;F1$
2260  INPUT HC$: INPUT T: INPUT Y$: INPUT Z$: INPUT NL: INPUT L%: INPUT M1
2270  DIM LN$(NL),QP%(NL),TC%(NL)
2280  DIM C1%(M1),C2%(M1),P%(M1)
2290  FOR J = 1 TO NL: INPUT LN$(J): NEXT J: INPUT ZA
2340  FOR J = 1 TO NL: INPUT QP%(J): INPUT TC%(J): NEXT J
2350  INPUT CD$: INPUT PM
2351  PRINT CL$
2352  PRINT OP$;F2$;R2$;DV$
2353  FOR J = 1 TO L%
2354  PRINT RD$;F2$;",R";J
2355  INPUT C1%(J): INPUT C2%(J): INPUT P%(J)
2356  NEXT J
2360  PRINT CL$: RETURN
2370  REM  EDIT A RECORD
2380  GOSUB 2090: IF G$ = "" THEN  INPUT "WHAT QSO# DO YOU WANT TO EDIT? ";G$
2390 X2 =  INT ( VAL (G$))
2400  IF X2 > 0 THEN  IF X2 < T THEN 2420
2410  GOTO 720
2420  HOME
2430  PRINT OP$;F3$;R3$;DV$
2431  PRINT RD$;F3$;",R";X2
2432  INPUT G$
2433 Y$(2) =  LEFT$ (G$,10):Y$(3) =  MID$ (G$,11,6):Y$(4) =  MID$ (G$,17,12)
2434 Y$(5) =  MID$ (G$,29,7):Y$(6) =  MID$ (G$,43,5):Y$(7) =  RIGHT$ (G$,10)
2435  FOR J = 2 TO 7:GS$ = Y$(J):LL =  LEN (GS$): GOSUB 2080:Y$(J) = G$:Y$(J + 10)
 = G$: NEXT J
2436  PRINT CL$
2440  GOSUB 2770
2442  FOR J = 2 TO 7: VTAB (J + 1): HTAB (14): PRINT Y$(J): NEXT J
2450  VTAB (11): INPUT "CHANGE THIS RECORD (Y/N)? ";A$
2460  IF A$ <  > "Y" THEN  HOME : GOTO 510
2470 LH% = 0
2480  FOR J = 1 TO NL: IF Y$(5) = LN$(J) THEN 2500
2490  NEXT J
2500 LX = LG:LY = J:LK = LY
2510 LB$ =  RIGHT$ (Y$(16),1)
2520  VTAB (10): INVERSE
2522  PRINT "              INSTRUCTIONS                "
2530  PRINT "                                          "
2535  PRINT " HIT <CR> TO LEAVE AN ITEM UNCHANGED.     "
2540  PRINT " TO CHANGE, ENTER THE NEW VALUE <CR>.     "
2545  PRINT " TO DELETE THIS ENTRY, CHANGE THE CALL    "
2550  PRINT "       TO '*'.                            "
2555  NORMAL
2560  FOR J = 2 TO 7
2570  VTAB (J + 1): HTAB (14): INPUT "";G$: IF G$ <  > "" THEN Y$(J) = G$: INVERSE

2580  VTAB (J + 1): HTAB (14): PRINT Y$(J): NORMAL
2590  NEXT J
2620  VTAB (17): INPUT "OK (Y/N)? ";A$:X1% = 4: PRINT
2630  IF A$ <  > "Y" THEN  HOME : GOSUB 700: GOTO 510
2640 CA$ = Y$(6): GOSUB 3520
```

```
2650  IF Y$(5) <  > Y$(15) THEN  GOSUB 2930: GOTO 2690: REM  BAND CHANGE
2660 LG = LK
2670  IF Y$(4) <  > Y$(14) THEN  GOSUB 2840: GOSUB 2870: GOSUB 3000: GOSUB 3040: G
OTO 2690: REM  CALL CHANGE
2680  IF Y$(6) <  > Y$(16) THEN  GOSUB 3000: GOSUB 3040
2690 LG = LX
2700  ON X1% GOTO 640,3080,3080,2710,2710,3090
2710  PRINT OP$;F3$;R3$;DV$
2720  PRINT WR$;F3$;",R";X2
2730  PRINT Y$(2);
2731  PRINT  SPC( 10 -  LEN (Y$(2)))Y$(3);
2732  PRINT  SPC( 6 -  LEN (Y$(3)))Y$(4);
2733  PRINT  SPC( 12 -  LEN (Y$(4)))Y$(5);
2734  PRINT  SPC( 14 -  LEN (Y$(5)))Y$(6);
2735  PRINT  SPC( 5 -  LEN (Y$(6)))Y$(7);
2737  PRINT  SPC( 10 -  LEN (Y$(7)))""
2739  PRINT CL$
2740  IF HC$ <  > "Y" THEN 2760
2750  PRINT LP$: PRINT "  ** EDIT TO REC #";X2;: FOR J = 2 TO 7: PRINT "  ";Y$(J);
: NEXT J
2755  PRINT : PRINT SCN$
2760  PRINT : GOTO 510
2770  PRINT "EDITING RCD# ";X2: PRINT : PRINT  TAB( 5)"DATE:"
2780  PRINT  TAB( 5)"TIME:"
2790  PRINT  TAB( 5)"CALL:"
2800  PRINT  TAB( 5)"BAND:"
2810  PRINT  TAB( 5)"CLASS:"
2815  PRINT  TAB( 5)"SECTION:"
2820  RETURN
2830  REM  CALL CHANGED
2840  IF Y$(4) = "*" THEN  RETURN
2850 CS$ = Y$(4): GOSUB 3110: IF X1% < 4 THEN  RETURN : REM  CHECK IF NEW CALL IS
DUPE
2860  GOSUB 1070: RETURN : REM  ADD NEW CALL
2870  IF X1% < 4 THEN  RETURN
2880  IF Y$(14) = "*" THEN  RETURN
2890 CS$ = Y$(14)
2900  GOSUB 3110:C1%(H%) =  - 32001:C2%(H%) =  - 32000: REM  DELETE OLD CALL
2910 X1% = 4: RETURN
2920  REM  CHANGE BAND
2930  FOR LK = 1 TO NL: IF Y$(5) = LN$(LK) THEN 2960
2940  NEXT LK
2950 X1% = 6: RETURN
2960 LG = LK: GOSUB 2840: IF X1% < 4 THEN  RETURN
2970 LG = LY: GOSUB 2870
2980  GOSUB 3000:LG = LK: GOSUB 3040: RETURN
2990  REM  CHANGE EXCHANGE
3000  IF Y$(14) = "*" THEN  RETURN
3010  IF CD$ = "Y" THEN  IF LB$ = "D" THEN 3030
3020 TC%(LG) = TC%(LG) - 1
3030  RETURN
3040  IF Y$(4) = "*" THEN  RETURN
3050  IF CD$ = "Y" THEN  IF LC$ = "D" THEN 3070
3060 TC%(LG) = TC%(LG) + 1
3070  RETURN
3080  PRINT BL$;"INVALID CALL SIGN": GOTO 510
3090  PRINT BL$;"INVALID BAND": GOTO 510
3100  REM  DUPE CHECKER:X1%=1=DUPE: 2=QUERY: 3=INVALID CALL: 4,5=OK
3110 X1% = 4:E(5) = 0:E(6) = 0:K = 1: IF  LEN (CS$) > 10 THEN X1% = 3: RETURN
3120  IF  RIGHT$ (CS$,1) = "?" THEN CS$ =  LEFT$ (CS$, LEN (CS$) - 1):X1% = 2:CT$
= CS$
3130  IF  LEN (CS$) < 4 THEN X1% = 3: RETURN
3140  IF  MID$ (CS$,3,1) < "A" THEN 3170
3150  IF  MID$ (CS$,2,1) =  > "A" THEN X1% = 3: RETURN
3160 CS$ = " " + CS$:K = 2:E(1) = 0
3170  FOR J = K TO  LEN (CS$)
3180 W$ =  MID$ (CS$,J,1): IF W$ = "/" THEN 3230
3190 E(J) =  ASC (W$) - AA: IF E(J) > 26 THEN X1% = 3: RETURN
```

```
3200  IF W$ < "0" THEN X1% = 3: RETURN
3210  IF E(J) < 1 THEN E(J) = E(J) + A0: IF E(J) > 36 THEN X1% = 3: RETURN
3220  NEXT J
3230  IF J > 7 THEN X1% = 3: RETURN
3240  IF J - K < 4 THEN X1% = 3: RETURN
3250  FOR J = 4 TO 6: IF E(J) > 26 THEN X1% = 3: RETURN
3260  NEXT J
3270 E1% =  - 32000 + E(1) * 1369 + E(2) * 37 + E(3)
3280 E2% =  - 32000 + E(4) * 1369 + E(5) * 37 + E(6)
3290  IF X1% = 2 THEN  RETURN
3300 H% = E(4) + LG * 26 - 26
3310  REM  SEARCH DUPE STRING
3320  IF C1%(H%) =  - 32000 THEN X1% = 4: RETURN
3330  IF C2%(H%) <  > E2% THEN 3350
3340  IF C1%(H%) = E1% THEN X1% = 1: RETURN
3350  IF P%(H%) = 0 THEN X1% = 5: RETURN
3360 H% = P%(H%): GOTO 3330
3370  REM  SCORE
3380  HOME
3385  PRINT : PRINT  SPC( 14)"SCORE SUMMARY": PRINT
3390  PRINT T1$
3400  GOSUB 3600
3450  PRINT T2$
3460  PRINT "TOTAL     ";X4; SPC( 16 -  LEN ( STR$ (X4)))X2
3470  PRINT : PRINT "SCORE = POWER MULT (";PM;") X ";X2;" = ";X2 * PM
3480  PRINT : RETURN
3520  IF  LEN (CA$) =  > 2 THEN  IF  LEN (CA$) <  = 3 THEN 3540
3530  PRINT BL$;"INVALID CLASS (OR REPORT). RE-ENTER? ";: GOSUB 2060:CA$ = G$: IF
X1% = 1 THEN  RETURN
3535  GOTO 3520
3540 LC$ =  LEFT$ (CA$,1): IF LC$ <  = "9" THEN  IF LC$ >  = "0" THEN 3560
3550  GOTO 3530
3560 LC$ =  RIGHT$ (CA$,1): IF LC$ >  = "A" THEN  IF LC$ > "E" THEN 3530
3570 LH% = 0: RETURN
3600 X2 = 0:X3 = 0:X4 = 0
3610  FOR J = 1 TO NL
3620 X4 = X4 + TC%(J):X1 = TC%(J) * QP%(J):X2 = X2 + X1
3630  PRINT LN$(J);
3640  PRINT  SPC( 10 -  LEN (LN$(J)))TC%(J);
3650  PRINT  SPC( 9 -  LEN ( STR$ (TC%(J))))QP%(J);
3660  PRINT  SPC( 7 -  LEN ( STR$ (QP%(J))))X1
3670  NEXT J
3680  RETURN
3900  PRINT DTE$;
3910  PRINT  SPC( 10 -  LEN (DTE$))UTC$;
3920  PRINT  SPC( 6 -  LEN (UTC$))CT$;
3930  PRINT  SPC( 12 -  LEN (CT$))LN$(LG);
3940  PRINT  SPC( 7 -  LEN (LN$(LG)))MD$;
3950  PRINT  SPC( 7 -  LEN (MD$))CA$;
3960  PRINT  SPC( 5 -  LEN (CA$))SK$;
3990  PRINT  SPC( 10 -  LEN (SK$))""
3999  RETURN
```

The Sweepstakes Logger—Operating Notes

ARRL Sweepstakes is another very popular contest with some unique scoring features. The SS LOGGER is tailored to those special features.

The SS LOGGER asks you a series of questions designed to determine your Sweepstakes exchange, which is based on a consecutive serial number, your power level ("precedence"), the year you were first

licensed ("check"), and ARRL section. After you answer those questions, plus the standard ones about the use of a real-time clock, hard-copy printout, etc., the SS LOGGER immediately comes up ready for QSO#1. It doesn't worry about bands (except to keep track of the one you tell it you're on), and it knows how many points QSOs are worth.

One of the special features of the SS LOGGER is that it has a data table including the official abbreviations for all ARRL sections. This enables the program to include a "clean sweep checker," which we discuss shortly. It also keeps you from getting an erroneous multiplier total by entering nonexistent sections. If you enter anything except an official section abbreviation during the multiplier portion of the exchange, you get an error message and an invitation to try again. Suggestion: put the list of official abbreviations on the wall and consult it until you have them memorized, which usually doesn't take long in SS. If you prefer other abbreviations over the official ones, all you have to do is change the DATA statements at the end of the SS LOGGER program to use your own abbreviations.

Because Sweepstakes has a lengthy exchange, the exchange you're supposed to send for each contact is displayed on the screen, complete with a reminder of the serial number for that particular contact. The received exchange can either be placed on the same line with the call sign, with spaces separating the number, precedence, check, and section (this is the fast way to do it), or you can enter these items one at a time. The SS LOGGER will prompt you for each item of the received exchange, lest you forget something. Don't forget to put a space between the received serial number and precedence; some operators treat the two as if they were a single item.

All of the normal menu commands work as they do on the other loggers, with one exception. The "STAT" command serves as the "clean sweep" checker in Sweepstakes. It lists all of the 74 multipliers by call area, flagging the ones you already have. This lets you see at a glance which sections you still need, something quite useful once you're up around 60 or 65 sections worked.

Also, since you can work a given station only once during Sweepstakes regardless of the band, the band-by-band dupecheck feature is not needed. To run a dupecheck, just type the call. If it's a dupe, the SS LOGGER will let you know.

For serious SS operators, contest timing is all-important. This is a contest in which it is vital that you know the accuracy of your real-time clock. The art of avoiding rubber clocking has become something of a specialty for some SS fans, and an inaccurate clock could get you in considerable trouble! When you hit that carriage return to log your starting time exactly at the instant the contest begins, you don't want to be using a clock that is a minute slow. And at the ending time, your clock can't be a minute fast, either.

SWEEPSTAKES LOGGER (TRS-80 and CP/M)

```
10 REM SWEEPSTAKES LOGGER (TRS-80 VERSION)
20 REM BY KC6A AND N6NB
30 REM (C) 1983 BY HAYDEN BOOK CO.,INC.
40 CLEAR 1000:BL$=CHR$(7):REM BL$ IS NEEDED; DON'T DELETE
50 CLS:HI$="":LO$=""
70 FA$="DUPELOGA":FB$="DUPELOGB":FC$="CONTSTLG"
80 ET=0:ZA=0:AA=ASC("A")-1:A0=ASC("A")-ASC("0")+26
90 T1$="          QSO'S X Q-PTS = POINTS  SECTIONS"
100 T2$="          -----         ------    ---"
110 GOSUB 2270
120 READ NS:DIM S$(NS),SC%(NS),SB(11),U1$(15),U2$(15),U3(15)
130 FOR J=0 TO 10:READ SB(J):NEXT J
140 FOR J=1 TO NS:READ S$(J):SC%(J)=0:NEXT J
150 PRINT:A$="":INPUT "NEW CONTEST (N) OR RESUME (R) ";A$
152 CLS
153 INPUT "WANT TO USE THE REAL TIME CLOCK TO LOG TIME";RC$
154 IF RC$<>"Y" THEN 160
155 INPUT "ENTER THE DATE (FORMAT:MM,DD,YY)";M5,D5,Y5
156 INPUT "ENTER THE TIME IN UTC (FORMAT:HH,MM)";HH,MM
157 POKE 16924,M5:POKE 16923,D5:POKE 16922,Y5:POKE 16921,HH:POKE 16920,MM
160 IF A$="N" THEN  220
170 IF A$<>"R" THEN PRINT BL$:GOTO 110
180 GOSUB 2490:REM READ IN FILE
190 IF HC$="Y" THEN GOSUB 1090
200 SS=27:GOTO 380
210 REM NEW CONTEST
220 PRINT:HC$="":INPUT "WANT HARD COPY WHILE LOGGING (Y/N) ";HC$
240 Y$="ARRL SWEEPSTAKES"
250 PRINT:PS$="":INPUT "ARE YOU USING LESS THAN 200 WATTS INPUT (Y/N) ";PS$
260 IF PS$="Y" THEN EX$=" A " ELSE EX$=" B "
270 PRINT:PS$="":INPUT "WHAT IS YOUR CALL SIGN ";PS$
280 EX$=EX$+PS$
290 PRINT:PS$="":INPUT "ENTER THE LAST TWO DIGITS OF THE YEAR YOU WERE FIRST LIC
    ENSED ";PS$:IF LEN(PS$)<>2 THEN 290
300 EX$=EX$+" "+PS$
310 PRINT:INPUT "WHAT IS YOUR ARRL SECTION ABBREVIATION ";SK$:GOSUB 3620:IF BS=1
    THEN 310
320 EX$=EX$+" "+SK$
330 M1=INT((FRE(0)-2000)/6)
340 DIM C1%(M1),C2%(M1),P%(M1)
350 T=1:L%=26:SS=L%+1:TC%=0:TS%=0
360 PRINT:PRINT "SPACE IS AVAILABLE FOR APPROXIMATELY";M1;"CALL SIGNS. GOOD LUCK
    !"
370 FOR J=1 TO L%:C1%(J)=-32000:P%(J)=0:NEXT J
380 GOSUB 3580
390 LH%=0:LP=1:LL=0:QP%=2
400 GOSUB 1150:IF A$="N" THEN Z$=DTE$:REM DATE
410 IF HC$="Y" THEN GOSUB 1110
420 GOSUB 1580:REM BAND
430 IF MD$="" THEN GOSUB 1200:REM MODE
440 GOSUB 1390:REM MENU
442 ZB$="":PRINT:INPUT "<ENTER> WHEN READY TO LOG STARTING TIME";ZB$
443 ZA=ZA+1:IF RC$<>"Y" THEN 446
444 GOSUB 1182:U1$(ZA)=UTC$:GOTO 450
446 INPUT "ENTER THE TIME IN UTC (FORMAT: 0000)";U1$(ZA)
450 PRINT:PRINT "LOGGER READY FOR QSO #";T;"-";LN$;MD$;TAB(45)T;EX$
460 PRINT "COMMAND OR CALL:";
470 X1%=0:GOSUB 2320:IF X1%=1 THEN 450
480 IF G$="?" THEN GOSUB 1390:GOTO 450
490 IF G$="QSY" THEN GOSUB 1580:GOSUB 1630:GOTO 450
500 IF G$="DATE" THEN GOSUB 1150:GOTO 450
510 IF G$="QRT" THEN GOSUB 3900:GOSUB 3510:GOSUB 1830:GOTO 2000
515 IF G$="QRX" THEN GOSUB 3900:GOSUB 1840:CLOSE:GOSUB 3580:GOTO 442
520 IF G$="SAVE" THEN GOSUB 1840:CLOSE:GOSUB 3580:GOTO 450
```

```
525 IF G$="RATE" THEN GOSUB 3900:ET=E7:GOTO 450
530 IF G$="SEC" THEN GOSUB 1630:GOSUB 1710:GOTO 450
540 IF G$="SCORE" THEN GOSUB 3510:GOTO 450
550 IF G$="EDIT" THEN GOTO 2670
560 IF G$="X" THEN GOSUB 1250:GOTO 450
570 GOTO 680
580 PRINT BL$:HI$
590 PRINT STRING$(38,140)
600 PRINT "**  IT'S A DUPE  **  BOO  **  HISS  **"
610 PRINT STRING$(38,131);LO$
620 GOTO 450
630 PRINT "INVALID ENTRY. PLEASE TRY AGAIN.";BL$:GOTO 450
640 PRINT "ABORTED":RETURN
650 PRINT "CAN'T DELETE";BL$:RETURN
660 PRINT "INVALID QSO #";BL$:GOTO 450
670 REM PROCESS A CALL SIGN
680 CS$=G$:CT$=G$
690 GOSUB 3240:ON X1% GOTO 580,630,630,700,700:REM DECODE CALL & JUMP ON RESULT
700 GOSUB 2350:IF LEN(G$)<>0 THEN 740
710 PRINT " -OK- ";
720 PRINT "SERIAL # (<CR> TO ABORT)? ";:GOSUB 2320:IF X1%=1 THEN 930
730 IF G$="" THEN 450
740 SN$=G$:IF LEN(SN$)>5 THEN PRINT BL$;"SERIAL # TOO LONG":PRINT:GOTO 720
750 GOSUB 2350
760 PC$=G$:IF X1%=1 THEN 930
770 IF PC$<>"A" THEN IF PC$<>"B" THEN PRINT "PRECEDENCE? ";:GOSUB 2320:GOTO 760
780 GOSUB 2350
790 CK$=G$:IF X1%=1 THEN 930
800 IF LEN(CK$)=2 THEN 820
810 PRINT "CHECK? ";:GOSUB 2320:GOTO 790
820 GOSUB 2350:IF G$="" THEN 850
830 SK$=G$:IF X1%=1 THEN 930
840 GOSUB 3620:IF BS=0 THEN 860
850 PRINT "SECTION? ";:GOSUB 2320:GOTO 830
860 IF RC$="Y" THEN GOSUB 1182:GOTO 870
862 GOSUB 2350:UTC$=G$:IF UTC$="" THEN PRINT "UTC? ";:GOSUB 2320:UTC$=G$:IF X1%=
    1 THEN 450
870 RN=T:GOSUB 950:REM WRITE LOG TO DISK
880 IF HC$="Y" THEN GOSUB 1010:REM WRITE TO PRINTER
890 LH%=H%:GOSUB 1040:REM ENTER IN DUPE SHEET
900 GOSUB 1070:REM ENTER SECTION
910 TC%=TC%+1
920 T=T+1:GOTO 450
930 CT$="*":RN=T:GOSUB 950:LH%=0:GOTO 920
940 REM WRITE LOG TO DISK
950 PRINT "...LOGGING"
960 LSET D1$=DTE$:LSET TM$=UTC$:LSET KL$=CT$
970 LSET B1$=LN$:LSET M1$=MD$:LSET SR$=SN$:LSET P$=PC$:LSET CH$=CK$:LSET SE$=SK$
980 PUT 3,RN
990 RETURN
1000 REM WRITE LOG TO PRINTER
1010 LPRINT T;TAB(6)DTE$;TAB(17)UTC$;TAB(24)LN$;TAB(30)MD$;TAB(36)CT$;TAB(48)SR$
    ;P$;CH$;SE$
1020 RETURN
1030 REM ENTER IN DUPE LIST
1040 IF X1%=5 THEN L%=L%+1:P%(H%)=L%:H%=L%
1050 C1%(H%)=E1%:C2%(H%)=E2%:P%(H%)=0:RETURN
1060 REM UPDATE SECTION
1070 IF SC%(SN)=0 THEN PRINT " * NEW SECTION *":TS%=TS%+1
1080 SC%(SN)=SC%(SN)+1:RETURN
1090 REM PRINTER INIT, IF USED
1100 RETURN
1110 LPRINT:LPRINT:LPRINT TAB(5)"CONTEST: ";Y$
1120 LPRINT TAB(5)"DATE:    ";Z$
1130 LPRINT:RETURN
1140 REM SET DATE
1150 IF RC$="Y" THEN 1182
1152 GOSUB 2350
```

```
1160 IF G$="" THEN PRINT:INPUT "WHAT IS THE DATE (MM/DD/YY) ";G$
1170 IF LEN(G$)>8 THEN PRINT "DATE TOO LONG";BL$:G$="":GOTO 1160
1180 DTE$=G$:RETURN
1182 DTE$=LEFT$(TIME$,8):UTC$=MID$(TIME$,10,2)+MID$(TIME$,13,2)
1184 PRINT "UTC=";UTC$:RETURN
1190 REM SET MODE
1200 GOSUB 2350
1210 IF G$="" THEN PRINT:INPUT "WHAT IS THE MODE ('CW' OR 'PH') ";G$
1220 IF LEN(G$)>3 THEN PRINT "INVALID MODE. 3 CHARACTERS MAX.";BL$:G$="":GOTO 12
     10
1230 MD$=G$:RETURN
1240 REM X DELETES LAST ENTRY
1250 IF LH%=0 THEN 650
1260 IF LH%<SS THEN IF P%(LH%)=0 THEN C1%(LH%)=-32000:GOTO 1280
1270 P%(LH%)=0:L%=L%-1
1280 TC%=TC%-1:LH%=0
1290 GOSUB 1340
1300 RN=T-1:CT$="*":GOSUB 950
1310 IF HC$="Y" THEN LPRINT "-- PREVIOUS CALL DELETED --"
1320 PRINT " DELETED":RETURN
1330 REM DELETE SECTION
1340 FOR J=1 TO NS:IF SK$=S$(J) THEN 1360
1350 NEXT J
1360 SC%(J)=SC%(J)-1:IF SC%(J)>0 THEN RETURN
1370 TS%=TS%-1:RETURN
1380 REM MENU
1390 CLS:PRINT HI$
1400 PRINT "       ARRL SWEEPSTAKES LOGGER"
1410 PRINT TAB(8)STRING$(23,131)
1420 PRINT " '?' DISPLAYS MENU"
1430 PRINT " 'QSY <BAND>' CHANGES BAND"
1440 PRINT " 'DATE <MM/DD/YY>' CHANGES DATE"
1450 PRINT " 'QRT' SAVES ALL CALLS AND EXITS"
1455 PRINT " 'QRX' SAVES ALL CALLS AND STOPS TIMER"
1460 PRINT " 'SAVE' SAVES AND RESUMES (ALLOW 2-3 MIN)"
1465 PRINT " 'RATE' SHOWS TOTAL TIME USED AND RATE"
1480 PRINT " 'SEC' SHOWS SECTIONS STILL NEEDED"
1490 PRINT " 'SCORE' GIVES RUNNING TOTAL SCORE"
1500 PRINT " 'EDIT <QSO #>' EDITS ANY LOGGED ENTRY"
1510 PRINT " 'X' DELETES THE LAST ENTRY"
1520 PRINT " '*' ABORTS AN ENTRY"
1525 CR$="":INPUT " <ENTER> TO CONTINUE ";CR$
1526 CLS:PRINT "          MENU (CONTINUED)"
1527 PRINT TAB(9)STRING$(16,131):PRINT
1530 PRINT " ENTRY FORMAT: 'CALL SERIAL# PRECEDENCE"
1535 PRINT "     CHECK SECTION <UTC>'"
1540 PRINT " ENTER CALL ONLY FOR DUPE CHECK"
1550 PRINT " <> DENOTES OPTIONAL ENTRIES"
1560 PRINT LO$:RETURN
1570 REM QSY
1580 GOSUB 2350
1590 IF G$="" THEN PRINT:INPUT "WHAT BAND DO YOU WANT ";G$
1600 LN$=G$:IF HC$="Y" THEN GOSUB 2440
1610 RETURN
1620 REM STATUS
1630 CLS:PRINT:PRINT "DATE= ";DTE$;
1640 PRINT "  BAND= ";LN$;
1650 PRINT "  MODE= ";MD$
1660 PRINT " SECTIONS WORKED= ";TS%;
1670 PRINT "  CONTACTS= ";TC%
1680 PRINT:RETURN
1690 REM LIST SECTIONS
1700 CLS:PRINT
1710 PRINT TAB(15)"------- SECTION SUMMARY -------"
1720 PRINT "ZONE     '*' INDICATES A SECTION WORKED"
1730 JJ=1
1740 FOR J=0 TO 10
1750 TB=4
```

```
1760 IF J=10 THEN PRINT "VE"; ELSE PRINT J;
1770 FOR K=1 TO SB(J)
1780 PRINT TAB(TB);:IF SC%(JJ)>0 THEN PRINT "*"; ELSE PRINT " ";
1790 PRINT S$(JJ);:TB=TB+5:JJ=JJ+1
1800 NEXT K:PRINT
1810 NEXT J:RETURN
1820 REM QRT
1830 CLOSE 3
1840 OPEN "O",1,FA$
1850 OPEN "R",2,FB$,6
1860 FIELD 2,2 AS C1$,2 AS C2$,2 AS P$
1870 PRINT#1,HC$:PRINT#1,T:PRINT#1,Y$:PRINT#1,Z$:PRINT#1,1:PRINT#1,L%:PRINT#1,M1
1880 PRINT#1," "
1885 PRINT#1,ZA:PRINT#1,ET:FOR J=1 TO ZA:PRINT#1,U1$(J):PRINT#1,U2$(J):PRINT#1,U
     3(J):NEXT J
1890 FOR J=1 TO L%
1900 LSET C1$=MKI$(C1%(J)):LSET C2$=MKI$(C2%(J)):LSET P$=MKI$(P%(J))
1910 PUT 2,J:NEXT J:CLOSE 2
1920 PRINT#1,QP%;",";TC%;",";TS%
1930 FOR K=1 TO NS
1940 PRINT#1,SC%(K)
1950 NEXT K
1960 PRINT#1,EX$
1970 PRINT#1,MD$
1980 RETURN
1990 REM WRITE SECT SUMMARY TO #1
2000 PRINT#1,BL$:REM NEW PAGE
2010 PRINT#1,TAB(28)"* SECTION SUMMARY *"
2020 PRINT#1," ":PRINT#1," ":PRINT#1," "
2030 FOR K=1 TO 5:PRINT#1,"  SECTION/QSOS ";:NEXT K:PRINT#1,
2040 TB=9
2050 FOR K=1 TO NS
2060 IF SC%(K)=0 THEN 2090
2070 PRINT#1,TAB(TB-LEN(S$(K)))S$(K);"/";SC%(K);
2080 TB=TB+15:IF TB>74 THEN TB=9:PRINT#1,
2090 NEXT K:PRINT#1," "
2100 REM WRITE SCORE TO LOG#1
2110 PRINT #1,BL$:REM NEW PAGE
2120 PRINT#1,TAB(14)"SCORE SUMMARY":PRINT#1,:PRINT#1,T1$
2130 IF HC$="Y" THEN LPRINT:LPRINT TAB(14)"SCORE SUMMARY":LPRINT:LPRINT T1$
2140 X1=QP%*TC%
2150 PRINT#1,TAB(11)TC%;TAB(20)QP%;TAB(27)X1;TAB(37)TS%
2160 IF HC$="Y" THEN LPRINT TAB(11)TC%;TAB(20)QP%;TAB(27)X1;TAB(37)TS%
2170 PRINT#1,T2$
2180 IF HC$="Y" THEN LPRINT T2$
2190 PRINT#1,TAB(25)"***********"
2200 PRINT#1,TAB(25)"*         *"
2210 PRINT#1,"SCORE = ";X1;" X ";TS%;TAB(25)"* ";X1*TS%;TAB(35)"*"
2220 PRINT#1,TAB(25)"*         *"
2230 PRINT#1,TAB(25)"***********"
2240 CLOSE
2250 IF HC$="Y" THEN LPRINT "SCORE = ";X1*TS%
2260 END
2270 CLS:PRINT
2280 PRINT CHR$(23);"ARRL SWEEPSTAKES LOGGING PROGRAM"
2285 PRINT:PRINT TAB(8)"BY KC6A AND N6NB"
2290 PRINT "(C) 1983 BY HAYDEN BOOK CO.,INC."
2300 :PRINT:PRINT:PRINT:RETURN
2310 REM INPUT SCAN ROUTINE
2320 GS$="":LINE INPUT GS$:LL=LEN(GS$)
2330 IF INSTR(GS$,"*")>0 THEN X1%=1:GOTO 640
2340 LP=1
2350 IF LP>LL THEN 2380:REM 2ND ENTRY
2360 FOR LT=LP TO LL:IF MID$(GS$,LT,1)<>" " THEN 2390
2370 NEXT LT
2380 G$="":RETURN
2390 LP=LT
2400 FOR LT=LP TO LL:IF MID$(GS$,LT,1)=" " THEN 2430
```

```
2410 NEXT LT
2420 LT=LT+1
2430 G$=MID$(GS$,LP,LT-LP):LP=LT:RETURN
2440 LPRINT
2450 LPRINT TAB(5)"QSY TO: ";LN$
2460 RETURN
2470 LL=LEN(GS$):GOTO 2340
2480 REM RESUME
2490 OPEN "I",1,FA$
2500 OPEN "R",2,FB$,6
2510 FIELD 2,2 AS C1$,2 AS C2$,2 AS P$
2520 INPUT#1,HC$:INPUT#1,T:INPUT#1,Y$:INPUT#1,Z$:INPUT#1,NL:INPUT#1,L%:INPUT#1,M
     1
2530 DIM C1%(M1),C2%(M1),P%(M1)
2540 INPUT#1,LN$
2545 INPUT#1,ZA:INPUT#1,ET:FOR J=1 TO ZA:INPUT#1,U1$(J):INPUT#1,U2$(J):INPUT#1,U
     3(J):NEXT J
2550 FOR J=1 TO L%
2560 GET 2,J
2570 C1%(J)=CVI(C1$):C2%(J)=CVI(C2$):P%(J)=CVI(P$)
2580 NEXT J
2590 INPUT#1,QP%,TC%,TS%
2600 FOR K=1 TO NS
2610 INPUT#1,SC%(K)
2620 NEXT K
2630 INPUT#1,EX$
2640 INPUT#1,MD$
2650 CLOSE:RETURN
2660 REM EDIT A RECORD
2670 GOSUB 2350:IF G$="" THEN INPUT "WHAT QSO# DO YOU WANT TO EDIT ";G$
2680 X2=INT(VAL(G$))
2690 PRINT "2675. TC=";TC%
2700 IF X2>0 THEN IF X2<T THEN 2720
2710 GOTO 660
2720 CLS
2730 GET 3,X2:Y$(1)=D1$:Y$(2)=TM$:Y$(3)=KL$:Y$(4)=B1$:Y$(5)=SR$:Y$(6)=P$:Y$(7)=C
     H$:Y$(8)=SE$
2740 GOSUB 3000
2750 PRINT:A$="":INPUT "CHANGE THIS RECORD (Y/N) ";A$
2760 IF A$<>"Y" THEN CLS:GOTO 450
2770 PRINT:PRINT TAB(12)"INSTRUCTIONS":PRINT
2780 PRINT "HIT <CR> TO LEAVE AN ITEM UNCHANGED."
2790 PRINT "TO CHANGE, ENTER NEW VALUE <CR>."
2800 PRINT "TO DELETE THIS CALL, CHANGE CALL TO '*'."
2810 PRINT:PRINT TAB(5)"DATE:";TAB(17)Y$(1);:A$="":INPUT A$:IF A$<>"" THEN Y$(1)
     =A$
2820 PRINT TAB(5)"TIME:";TAB(17)Y$(2);:A$="":INPUT A$:IF A$<>""THEN Y$(2)=A$
2830 PRINT TAB(5)"CALL:";TAB(17)Y$(3);:A$="":INPUT A$:IF A$<>""THEN Y$(3)=A$
2840 PRINT TAB(5)"BAND:";TAB(17)Y$(4);:A$="":INPUT A$:IF A$<>"" THEN Y$(4)=A$
2850 PRINT TAB(5)"SERIAL#:";TAB(17)Y$(5);:A$="":INPUT A$:IF A$<>"" THEN Y$(5)=A$
2860 PRINT TAB(5)"PRECEDENCE:";TAB(17)Y$(6);:A$="":INPUT A$:IF A$<>"" THEN Y$(6)
     =A$
2870 PRINT TAB(5)"CHECK:";TAB(17)Y$(7);:A$="":INPUT A$:IF A$<>"" THEN Y$(7)=A$
2880 PRINT TAB(5)"SECTION:";TAB(17)Y$(8);:A$="":INPUT A$:IF A$<>"" THEN Y$(8)=A$
2890 PRINT:A$="":INPUT "OK (Y/N) ";A$:X1%=4
2900 IF A$<>"Y" THEN CLS:GOSUB 640:GOTO 450
2910 IF Y$(3)<>KL$ THEN GOSUB 3070:GOSUB 3100
2920 IF Y$(8)<>SE$ THEN GOSUB 3160:GOSUB 3190:IF BS=1 THEN 450
2930 ON X1% GOTO 580,3210,3210,2940,2940,3220
2940 LSET D1$=Y$(1):LSET TM$=Y$(2):LSET KL$=Y$(3)
2950 LSET B1$=Y$(4):LSET SR$=Y$(5):LSET P$=Y$(6):LSET CH$=Y$(7):LSET SE$=Y$(8)
2960 PUT 3,X2
2970 IF HC$<>"Y" THEN 2990
2980 LPRINT "  ** EDIT ON REC #";X2;": ";D1$;" ";TM$;" ";KL$;" ";B1$;" ";M1$;" "
     ;SR$;P$;CH$;SE$
2990 GOTO 450
3000 PRINT:PRINT  TAB(5)"DATE:";TAB(14)Y$(1)
3010 PRINT TAB(5)"TIME:";TAB(14)Y$(2)
```

```
3020 PRINT TAB(5)"CALL:";TAB(14)Y$(3)
3030 PRINT TAB(5)"BAND:";TAB(14)Y$(4)
3040 PRINT TAB(5)"EXCHANGE:"TAB(14)Y$(5);Y$(6);;Y$(7);Y$(8)
3050 RETURN
3060 REM CALL CHANGED
3070 IF Y$(3)="*" THEN Y$(8)="*":RETURN
3080 GS$=Y$(3):GOSUB 2470:CS$=G$:GOSUB 3240:IF X1%<4 THEN RETURN:REM CHECK IF NE
     W CALL IS DUPE
3090 GOSUB 1040:TC%=TC%+1:RETURN
3100 IF X1%<4 THEN RETURN:IF LEFT$(KL$,1)="*" THEN RETURN
3110 IF LEFT$(KL$,1)="*" THEN RETURN
3120 GS$=KL$:GOSUB 2470:CS$=G$
3130 GOSUB 3240:C1%(H%)=-32001:C2%(H%)=-32000:REM DELETE OLD CALL
3140 TC%=TC%-1:X1%=4:RETURN
3150 REM EXCH CHANGE
3160 IF LEFT$(SE$,1)="*" THEN RETURN
3170 GS$=SE$:GOSUB 2470:SK$=G$
3180 GOTO 1340:REM DELETE OLD SECTION
3190 BS=0:IF Y$(8)="*" THEN RETURN
3200 GS$=Y$(8):GOSUB 2470:SK$=G$:GOSUB 3620:GOSUB 1070:RETURN:REM ADD NEW SECTIO
     N
3210 PRINT BL$;"INVALID CALL SIGN":GOTO 450
3220 PRINT BL$;"INVALID BAND":GOTO 450
3230 REM DUPE CHECKER: X1%=1=DUPE: 2=QUERY: 3=INVALID CALL: 4,5=OK
3240 X1%=4:E(5)=0:E(6)=0:K=1:IF LEN(CS$)>10 THEN X1%=3:RETURN
3250 IF RIGHT$(CS$,1)="?" THEN CS$=LEFT$(CS$,LEN(CS$)-1):X1%=2:CT$=CS$
3260 IF LEN(CS$)<4 THEN X1%=3:RETURN
3270 IF MID$(CS$,3,1)<"A" THEN 3300
3280 IF MID$(CS$,2,1)=>"A" THEN X1%=3:RETURN
3290 CS$=" "+CS$:K=2:E(1)=0
3300 FOR J=K TO LEN(CS$)
3310 W$=MID$(CS$,J,1):IF W$="/" THEN 3360
3320 E(J)=ASC(W$)-AA:IF E(J)>26 THEN X1%=3:RETURN
3330 IF W$<"0" THEN X1%=3:RETURN
3340 IF E(J)<1 THEN E(J)=E(J)+A0:IF E(J)>36 THEN X1%=3:RETURN
3350 NEXT J
3360 IF J>7 THEN X1%=3:RETURN
3370 IF J-K<4 THEN X1%=3:RETURN
3380 FOR J=4 TO 6:IF E(J)>26 THEN X1%=3:RETURN
3390 NEXT J
3400 E1%=-32000+E(1)*1369+E(2)*37+E(3)
3410 E2%=-32000+E(4)*1369+E(5)*37+E(6)
3420 IF X1%=2 THEN RETURN
3430 H%=E(4)
3440 REM SEARCH DUPE STRING
3450 IF C1%(H%)=-32000 THEN X1%=4:RETURN
3460 IF C2%(H%)<>E2% THEN 3480
3470 IF C1%(H%)=E1% THEN X1%=1:RETURN
3480 IF P%(H%)=0 THEN X1%=5:RETURN
3490 H%=P%(H%):GOTO 3460
3500 REM SCORE
3510 PRINT:PRINT TAB(22)"SCORE SUMMARY":PRINT
3520 PRINT T1$
3530 X2=0:X3=0:X4=0
3540 PRINT TAB(11)TC%;TAB(20)QP%;TAB(27)TC%*QP%;TAB(37)TS%
3550 PRINT T2$
3560 PRINT:PRINT TAB(10)"SCORE =";TC%*QP%*TS%
3570 PRINT:RETURN
3580 OPEN "R",3,FC$,47
3590 FIELD 3,8 AS D1$,4 AS TM$,10 AS KL$,5 AS B1$,5 AS M1$,5 AS SR$,2 AS P$,3 AS
     CH$,5 AS SE$
3600 RETURN
3610 REM SECTION CHECK
3620 FOR J=1 TO NS:IF SK$=S$(J) THEN 3650
3630 NEXT J
3640 PRINT BL$;"INVALID SECTION":BS=1:RETURN
3650 SN=J:BS=0:RETURN
3652 REM VALID SECTION LIST
```

```
3653 REM FIRST LINE IS TOTAL # OF SECTIONS FOLLOWED BY # SECTIONS IN EACH CALL A
     REA.
3654 REM EACH LINE AFTER THAT CONTAINS THE ARRL SECTION ABBREVIATIONS.
3660 DATA 74,8,7,5,4,10,7,10,9,3,3,8
3670 DATA "CO","IA","KS","MN","MO","NE","ND","SD":REM CALL ZONE "0"
3680 DATA "CT","EMA","ME","NH","RI","VT","WMA":REM "1'S"
3690 DATA "ENY","NLI","NNJ","SNJ","WNY":REM "2'S"
3700 DATA "EPA","DE","MDC","WPA":REM "3'S"
3710 DATA "AL","GA","KY","NC","NFL","SC","SFL","TN","VA","WIN":REM "4'S"
3720 DATA "AR","LA","MS","NM","NTX","OK","STX":REM "5'S"
3730 DATA "EB","LAX","ORG","SB","SCV","SDG","SF","SJV","SV","PAC":REM "6'S"
3740 DATA "AZ","ID","MT","NV","OR","UT","WA","WY","AK":REM "7'S"
3750 DATA "MI","OH","WV":REM "8'S"
3760 DATA "IL","IN","WI":REM  "9'S"
3770 DATA "MAR","PQ","ON","MB","SK","AB","BC","VE8":REM "VE'S"
3900 IF RC$<>"Y" THEN 3920
3910 GOSUB 1182:U2$(ZA)=UTC$:GOTO 3930
3920 PRINT:INPUT "ENTER THE TIME IN UTC";U2$(ZA)
3930 UX=VAL(U1$(ZA)):UY=VAL(U2$(ZA))
3932 U7=INT(UX/100):U4=((UX/100)-U7)/.6:U7=U7+U4
3940 U5=INT(UY/100):U6=((UY/100)-U5)/.6:U5=U5+U6
3950 IF U5<U7 THEN U5=U5+24
3960 E7=ET:ET=ET+U5-U7:U3(ZA)=T:IF ET=0 THEN 3990
3970 PRINT:PRINT "TIME USED SO FAR:";INT(ET);" HRS - ";INT((ET-INT(ET))*60+.5);"
     MINS"
3980 PRINT:PRINT "OVERALL RATE:  ";INT ((T-1)/ET+.5);" QSOS PER HOUR"
3990 RETURN
```

SWEEPSTAKES LOGGER (Apple)

```
10  REM  SWEEPSTAKES LOGGER (APPLE)
20  REM  BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
40 BL$ =  CHR$ (7): REM  BELL
50 D$ =  CHR$ (4):Q$ =  CHR$ (34)
52 LP$ = D$ + "PR#1":SCN$ = D$ + "PR#0"
54 RD$ = D$ + "READ ":WR$ = D$ + "WRITE "
56 CL$ = D$ + "CLOSE ":OP$ = D$ + "OPEN "
57 DL$ = D$ + "DELETE ":DV$ = ",D1"
70 F1$ = "DUPELOGA": HOME
72 F2$ = "DUPELOGB":R2$ = ",L19"
74 F3$ = "CONTSTLG":R3$ = ",L58"
80 AA =  ASC ("A") - 1:A0 =  ASC ("A") -  ASC ("0") + 26:ET = 0:ZA = 0
90 T1$ = "       QSO'S X Q-PTS = POINTS  SECTIONS"
100 T2$ = "       -----         ------    ---"
110  GOSUB 2270
115  PRINT : PRINT "ARE YOU USING A REAL TIME CLOCK FOR"
116  INPUT "LOGGING THE TIME (Y/N)? ";RC$
120  READ NS: DIM S$(NS),SC%(NS),SB(11),Y$(20),U1$(15),U2$(15),U3(15)
130  FOR J = 0 TO 10: READ SB(J): NEXT J
140  FOR J = 1 TO NS: READ S$(J):SC%(J) = 0
142  IF  LEN (S$(J)) > 4 THEN  PRINT BL$;"SECTION NAME ";: INVERSE : PRINT S$(J);:
 NORMAL : PRINT " IS TOO LONG.": END
145  NEXT J
150  PRINT : INPUT "NEW CONTEST (N) OR RESUME (R)? ";A$
160  IF A$ = "N" THEN 220
170  IF A$ <  > "R" THEN  PRINT BL$: GOTO 110
180  GOSUB 2490: REM  READ IN FILE
185  PRINT : PRINT "RESUMING ";Y$;" OF ";Z$
190  IF HC$ = "Y" THEN  GOSUB 1090
```

```
200 SS = 27: GOTO 390
210  REM  NEW CONTEST
220  PRINT : INPUT "WANT HARD COPY WHILE LOGGING (Y/N)? ";HC$
230  IF HC$ = "Y" THEN  INPUT "TURN ON PRINTER. <CR> WHEN READY? ";AA$: GOSUB 1090
240 Y$ = "SWEEPSTAKES CONTEST"
250  PRINT : PRINT "ARE YOU USING LESS THAN 200 WATTS": INPUT "     INPUT (Y/N)? ";
PS$
260 EX$ = " B ": IF PS$ = "Y" THEN EX$ = " A "
270  PRINT : INPUT "WHAT IS YOUR CALL SIGN? ";PS$
280 EX$ = EX$ + PS$
290  PRINT : PRINT "WHAT IS THE LAST TWO DIGITS OF THE": INPUT "      YEAR YOU WERE
 FIRST LICENSED? ";PS$
295  IF  LEN (PS$) <  > 2 THEN 290
300 EX$ = EX$ + " " + PS$
310  PRINT : INPUT "WHAT IS YOUR ARRL SECTION ABBREVIATION? ";SK$
312  GOSUB 3620: IF BS = 1 THEN  GOSUB 1710: GOTO 310
320 EX$ = EX$ + " " + SK$
330 M1 =  FRE (0): IF M1 < 0 THEN M1 = M1 + 65536
332 M1 =  INT ((M1 - 2000) / 6)
340  DIM C1%(M1),C2%(M1),P%(M1)
350 T = 1:L% = 26:SS = L% + 1:TC% = 0:TS% = 0:MD$ = ""
360  PRINT : PRINT "SPACE IS AVAILABLE FOR APPROXIMATELY": PRINT M1;" CALL SIGNS.
 GOOD LUCK!"
370  FOR J = 1 TO L%:C1%(J) =  - 32000:P%(J) = 0: NEXT J
390 LH% = 0:LP = 1:LL = 0:QP% = 2
400  GOSUB 1150: IF A$ = "N" THEN Z$ = DTE$: REM  DATE
410  IF HC$ = "Y" THEN  GOSUB 1110
420  GOSUB 1580: REM  BAND
430  IF MD$ = "" THEN  GOSUB 1200: REM  MODE
440  GOSUB 1390: REM  MENU
442  PRINT : INPUT "<CR> TO LOG STARTING TIME";ZB$
443 ZA = ZA + 1: IF RC$ <  > "Y" THEN 446
444  GOSUB 1182:U1$(ZA) = UTC$: GOTO 450
446  INPUT "ENTER THE TIME IN UTC";U1$(ZA)
450  PRINT : PRINT "LOGGER READY FOR QSO #";T;".  BAND: ";LN$;MD$: PRINT  TAB( 5)T
;EX$
460  PRINT "COMMAND OR CALL";
470 X1% = 0: GOSUB 2320: IF X1% = 1 THEN 450
480  IF G$ = "?" THEN  GOSUB 1390: GOTO 450
490  IF G$ = "QSY" THEN  GOSUB 1580: GOSUB 1630: GOTO 450
500  IF G$ = "DATE" THEN  GOSUB 1150: GOTO 450
510  IF G$ = "QRT" THEN  GOSUB 4000: GOSUB 3510: GOSUB 1830: GOSUB 2000: GOTO 2250
515  IF G$ = "QRX" THEN  GOSUB 4000: GOSUB 1830: GOSUB 2240: GOTO 442
520  IF G$ = "SAVE" THEN  GOSUB 1830: GOSUB 2240: GOTO 450
525  IF G$ = "RATE" THEN  GOSUB 4000:ET = E7: GOTO 450
530  IF G$ = "STAT" THEN  GOSUB 1630: GOSUB 1710: GOTO 450
540  IF G$ = "SCORE" THEN  GOSUB 3510: GOTO 450
550  IF G$ = "EDIT" THEN  GOTO 2670
560  IF G$ = "X" THEN  GOSUB 1250: GOTO 450
570  GOTO 680
580  PRINT BL$: FLASH
590  PRINT "****************************************"
600  PRINT "**  IT'S A DUPE  **  BOO  **  HISS  **"
610  PRINT "****************************************"
620  NORMAL : GOTO 450
630  PRINT "INVALID ENTRY. PLEASE TRY AGAIN.";BL$: GOTO 450
640  PRINT "ABORTED": RETURN
650  PRINT "CAN'T DELETE";BL$: RETURN
660  PRINT "INVALID QSO#";BL$: GOTO 450
670  REM  PROCESS A CALL SIGN
680 CS$ = G$:CT$ = G$
690  GOSUB 3240: ON X1% GOTO 580,630,630,700,700: REM  DECODE CALL & JUMP ON RESUL
T
700  GOSUB 2350: IF  LEN (G$) <  > 0 THEN 740
710  PRINT " -OK- ";
720  PRINT "SERIAL # ";: GOSUB 2320: IF X1% = 1 THEN 450
730  IF G$ = "" THEN 720
740 SN$ = G$: IF  LEN (SN$) > 5 THEN  PRINT BL$;"SERIAL # TOO LONG": PRINT : GOTO
720
```

```
750  GOSUB 2350
760 PC$ = G$: IF X1% = 1 THEN 930
770  IF PC$ <  > "A" THEN  IF PC$ <  > "B" THEN  PRINT "PRECEDENCE ";: GOSUB 2320:
 GOTO 760
780  GOSUB 2350
790 CK$ = G$: IF X1% = 1 THEN 930
800  IF  LEN (CK$) = 2 THEN 820
810  PRINT "CHECK ";: GOSUB 2320: GOTO 790
820  GOSUB 2350: IF G$ = "" THEN 850
830 SK$ = G$: IF X1% = 1 THEN 930
840  GOSUB 3620: IF BS = 0 THEN 860
850  PRINT "SECTION ";: GOSUB 2320: GOTO 830
860  IF RC$ = "Y" THEN  GOSUB 1182: GOTO 870
862  GOSUB 2350:UTC$ = G$: IF UTC$ = "" THEN  PRINT "UTC ";: GOSUB 2320:UTC$ = G$:
 IF X1% = 1 THEN 450
865  IF  LEN (UTC$) > 4 THEN  PRINT "INVALID UTC";BL$: GOTO 860
870 RN = T: GOSUB 950: REM  WRITE LOG TO DISK
880  IF HC$ = "Y" THEN  GOSUB 1010: REM  WRITE TO PRINTER
890 LH% = H%: GOSUB 1040: REM  ENTER IN DUPE SHEET
900  GOSUB 1070: REM  ENTER SECTION
910 TC% = TC% + 1
920 T = T + 1: GOTO 450
930 CT$ = "*":RN = T: GOSUB 950:LH% = 0: GOTO 920: REM  ABORTED ENTRY
940  REM  WRITE LOG TO DISK
950  PRINT "..LOGGING"
960  PRINT OP$;F3$;R3$;DV$
970  PRINT WR$;F3$;",R";RN
980  GOSUB 3900
982  PRINT CL$
990  RETURN
1000  REM  WRITE LOG TO PRINTER
1010  PRINT LP$: PRINT T; SPC( 5 -  LEN ( STR$ (T)));: GOSUB 3900: PRINT SCN$
1020  RETURN
1030  REM  ENTER IN DUPE LIST
1040  IF X1% = 5 THEN L% = L% + 1:P%(H%) = L%:H% = L%
1050 C1%(H%) = E1%:C2%(H%) = E2%:P%(H%) = 0: RETURN
1060  REM  UPDATE SECTION
1070  IF SC%(SN) = 0 THEN  INVERSE : PRINT " * NEW SECTION *": NORMAL :TS% = TS% +
 1
1080 SC%(SN) = SC%(SN) + 1: RETURN
1090  REM  PRINTER INIT IF USED
1100  RETURN
1110  PRINT LP$: PRINT : PRINT : PRINT  SPC( 4)"CONTEST: ";Y$
1120  PRINT  SPC( 4)"DATE:     "Z$
1130  PRINT : PRINT SCN$: RETURN
1140  REM  SET DATE
1150  IF RC$ = "Y" THEN 1182
1152  GOSUB 2350
1160  IF G$ = "" THEN  PRINT : PRINT "WHAT IS THE DATE (MM/DD/YY) ";: GOSUB 2320:
GOTO 1160
1170  IF  LEN (G$) > 8 THEN  PRINT "DATE TOO LONG";BL$:G$ = "": GOTO 1160
1180 DTE$ = G$: RETURN
1182  PRINT D$;"IN#1": INPUT "DAY/DATE/TIME = ";G$: PRINT D$;"IN#0":DTE$ =  MID$ (
G$,3,8):UTC$ =  MID$ (G$,12,2) +  MID$ (G$,15,2)
1184  RETURN
1190  REM  SET MODE
1200  GOSUB 2350
1210  IF G$ = "" THEN  PRINT : INPUT "WHAT IS THE MODE ('CW' OR 'PH')? ";G$
1220  IF  LEN (G$) > 3 THEN  PRINT "INVALID MODE. 3 CHARACTERS MAX.";BL$:G$ = "":
GOTO 1210
1230 MD$ = G$: RETURN
1240  REM  X DELETES LAST ENTRY
1250  IF LH% = 0 THEN 650
1260  IF LH% < SS THEN  IF P%(LH%) = 0 THEN C1%(LH%) =  - 32000; GOTO 1280
1270 P%(LH%) = 0:L% = L% - 1
1280 TC% = TC% - 1:LH% = 0
1290  GOSUB 1340
1300 RN = T - 1:CT$ = "*": GOSUB 950
```

```
1310  IF HC$ = "Y" THEN  PRINT LP$: PRINT "-- PREVIOUS CALL DELETED --": PRINT SCN
$
1320  PRINT " DELETED": RETURN
1330  REM  DELETE SECTION
1340  FOR J = 1 TO NS: IF SK$ = S$(J) THEN 1360
1350  NEXT J
1360 SC%(J) = SC%(J) - 1: IF SC%(J) > 0 THEN  RETURN
1370 TS% = TS% - 1: RETURN
1380  REM  MENU
1390  HOME : PRINT : INVERSE
1400  PRINT "     SWEEPSTAKES CONTEST LOGGER MENU     "
1410  PRINT "                                         "
1420  PRINT " '?' DISPLAYS MENU                       "
1430  PRINT " 'QSY <BAND>' CHANGES BAND               "
1440  PRINT " 'DATE <MM/DD/YY>' CHANGES DATE          "
1450  PRINT " 'QRT' SAVES ALL CALLS AND EXITS         "
1455  PRINT " 'QRX' SAVES ALL CALLS AND STOPS TIMER   "
1460  PRINT " 'SAVE' SAVES ALL CALLS AND CONTINUES    "
1470  PRINT "         (NOTE: ALLOW 2-3 MIN FOR SAVE)  "
1475  PRINT " 'RATE' SHOWS TOTAL TIME USED AND RATE   "
1480  PRINT " 'STAT' DISPLAYS SECTIONS NEEDED         "
1490  PRINT " 'SCORE' GIVES RUNNING TOTAL SCORE       "
1500  PRINT " 'EDIT <QSO #>' EDITS ANY LOGGED ENTRY   "
1510  PRINT " 'X' DELETES THE LAST ENTRY              "
1520  PRINT " '-' ABORTS AN ENTRY                     "
1530  PRINT " ENTRY FORMAT: 'CALL SERIAL# PRECEDENCE"
1535  PRINT "                CHECK SECTION <UTC>'     "
1540  PRINT " ENTER 'CALL' ONLY FOR DUPE CHECK        "
1550  PRINT " <> DENOTES OPTIONAL ENTRIES             "
1560  NORMAL : RETURN
1570  REM  QSY
1580  GOSUB 2350
1590  IF G$ = "" THEN  PRINT : INPUT "WHAT BAND DO YOU WANT? ";G$
1595  IF  LEN (G$) > 5 THEN  PRINT BL$;"5 CHARACTERS MAX":G$ = "": GOTO 1590
1600 LN$ = G$: IF HC$ = "Y" THEN  GOSUB 2440
1610  RETURN
1620  REM  STATUS
1630  HOME : PRINT : PRINT "DATE= ";DTE$;
1640  PRINT "  BAND= ";LN$;
1650  PRINT "  MODE= ";MD$
1660  PRINT " SECTIONS WORKED= ";TS%;
1670  PRINT "  CONTACTS= ";TC%
1680  PRINT : RETURN
1690  REM  LIST SECTIONS
1700  HOME
1710  PRINT  TAB( 4)"------- SECTION SUMMARY -------"
1720  PRINT "ZONE       SECTIONS WORKED ARE ";: INVERSE : PRINT " INVERSE ": NORMA
L
1730 JJ = 1
1740  FOR J = 0 TO 10
1750 TB = 4
1760  IF J = 10 THEN  PRINT "VE";: GOTO 1770
1762  PRINT J;
1770  FOR K = 1 TO SB(J)
1775  IF TB > 35 THEN  PRINT :TB = 4
1780  PRINT  TAB( TB);: IF SC%(JJ) > 0 THEN  INVERSE
1790  PRINT S$(JJ);: NORMAL :TB = TB + 5:JJ = JJ + 1
1800  NEXT K: PRINT
1810  NEXT J: RETURN
1820  REM  QRT
1830  PRINT OP$;F1$;DV$
1840  PRINT DL$;F1$
1850  PRINT OP$;F1$
1860  PRINT WR$;F1$
1870  PRINT HC$: PRINT T: PRINT Y$: PRINT Z$: PRINT 1: PRINT L%: PRINT M1
1880  PRINT " ": PRINT ZA: PRINT ET: FOR J = 1 TO ZA: PRINT U1$(J): PRINT U2$(J):
PRINT U3(J): NEXT J
1920  PRINT QP%: PRINT TC%: PRINT TS%
```

```
1930  FOR K = 1 TO NS
1940  PRINT SC%(K)
1950  NEXT K
1960  PRINT EX$
1970  PRINT MD$
1980  RETURN
1990  REM  WRITE SECT SUMMARY
2000  PRINT BL$: REM  NEW PAGE
2010  PRINT Q$; SPC( 27)"* SECTION SUMMARY *";Q$
2020  PRINT : PRINT : PRINT
2030  PRINT Q$;: FOR K = 1 TO 5: PRINT "  SECTION/QSOS ";: NEXT K: PRINT Q$
2040 TB = 9
2050  FOR K = 1 TO NS
2060  IF SC%(K) = 0 THEN 2090
2065  IF TB = 9 THEN  PRINT Q$;
2070  PRINT  SPC( 9 -  LEN (S$(K)))S$(K);"/";SC%(K); SPC( 5 -  LEN ( STR$ (SC%(K))
));
2080 TB = TB + 15: IF TB > 74 THEN TB = 9: PRINT Q$
2090  NEXT K: IF TB > 9 THEN  PRINT Q$
2095  PRINT
2100  REM  WRITE SCORE TO LOG
2110  PRINT BL$: REM  NEW PAGE
2120  PRINT Q$; SPC( 13)"SCORE SUMMARY";Q$: PRINT : PRINT Q$;T1$;Q$
2140  GOSUB 3800
2170  PRINT Q$;T2$;Q$
2190  PRINT Q$; SPC( 24)"***********"
2200  PRINT Q$; SPC( 24)"*          *"
2210  PRINT "SCORE = ";X1;" X ";TS%; SPC( 13 -  LEN ( STR$ (X1)) -  LEN ( STR$ (TS
%)))"* ";
2212  PRINT X1 * TS%; SPC( 8 -  LEN ( STR$ (X1 * TS%)))"*"
2220  PRINT Q$; SPC( 24)"*          *"
2230  PRINT Q$; SPC( 24)"***********"
2235  PRINT "1": REM  EOF MARK
2240  PRINT CL$
2241  PRINT OP$;F2$;DV$
2242  PRINT DL$;F2$
2243  PRINT OP$;F2$;R2$
2244  FOR J = 1 TO L%
2245  PRINT WR$;F2$;",R";J
2246  PRINT C1%(J): PRINT C2%(J): PRINT P%(J)
2247  NEXT J
2248  PRINT CL$: RETURN
2250  IF HC$ = "Y" THEN  PRINT LP$: GOSUB 3510: PRINT SCN$
2260  END
2270  HOME : PRINT : INVERSE : REM  PRINT TITLE
2280  PRINT "  SWEEPSTAKES CONTEST LOGGING PROGRAM  "
2290  PRINT "           BY KC6A AND N6NB            "
2292  PRINT "   (C) 1983 BY HAYDEN BOOK CO., INC.   "
2300  NORMAL : PRINT : RETURN
2310  REM  INPUT SCAN ROUTINE
2320  INPUT GS$:LL =  LEN (GS$): IF LL = 0 THEN 2380
2330  FOR J = 1 TO LL: IF  MID$ (GS$,J,1) = "-" THEN X1% = 1: GOTO 640
2331  NEXT J
2340 LP = 1
2350  IF LP > LL THEN 2380: REM  2ND ENTRY
2360  FOR LT = LP TO LL: IF  MID$ (GS$,LT,1) <  > " " THEN 2390
2370  NEXT LT
2380 G$ = "": RETURN
2390 LP = LT
2400  FOR LT = LP TO LL: IF  MID$ (GS$,LT,1) = " " THEN 2430
2410  NEXT LT
2420 LT = LT + 1
2430 G$ =  MID$ (GS$,LP,LT - LP):LP = LT: RETURN
2440  PRINT LP$: PRINT
2450  PRINT  SPC( 4)"QSY TO: ";LN$
2460  PRINT SCN$: RETURN
2470 LL =  LEN (GS$): GOTO 2340
2480  REM  RESUME
```

```
2490  PRINT OP$;F1$;DV$
2500  PRINT RD$;F1$
2520  INPUT HC$: INPUT T: INPUT Y$: INPUT Z$: INPUT NL: INPUT L%: INPUT M1
2530  DIM C1%(M1),C2%(M1),P%(M1)
2540  INPUT LN$: INPUT ZA: INPUT ET: FOR J = 1 TO ZA: INPUT U1$(J): INPUT U2$(J):
INPUT U3(J): NEXT J
2590  INPUT QP%: INPUT TC%: INPUT TS%
2600  FOR K = 1 TO NS
2610  INPUT SC%(K)
2620  NEXT K
2630  INPUT EX$
2640  INPUT MD$
2641  PRINT CL$
2642  PRINT OP$;F2$;R2$;DV$
2643  FOR J = 1 TO L%
2644  PRINT RD$;F2$;",R";J
2645  INPUT C1%(J): INPUT C2%(J): INPUT P%(J)
2646  NEXT J
2650  PRINT CL$: RETURN
2660  REM  EDIT A RECORD
2670  GOSUB 2350: IF G$ = "" THEN  INPUT "WHAT QSO # DO YOU WANT TO EDIT? ";G$
2680 X2 =  INT ( VAL (G$))
2700  IF X2 > 0 THEN  IF X2 < T THEN 2720
2710  GOTO 660
2720  HOME
2730  PRINT OP$;F3$;R3$;DV$
2731  PRINT RD$;F3$;",R";X2
2732  INPUT G$
2733  PRINT CL$
2734 Y$(1) =  LEFT$ (G$,10):Y$(2) =  MID$ (G$,11,6)
2735 Y$(3) =  MID$ (G$,17,12):Y$(4) =  MID$ (G$,29,7)
2736 Y$(5) =  MID$ (G$,43,5):Y$(6) =  MID$ (G$,48,2)
2737 Y$(7) =  MID$ (G$,50,3):Y$(8) =  RIGHT$ (G$,5)
2738  FOR J = 1 TO 8:GS$ = Y$(J):LL =  LEN (GS$): GOSUB 2340:Y$(J) = G$:Y$(J + 10)
 = G$: NEXT J
2740  GOSUB 3000
2742  FOR J = 1 TO 8: VTAB (J + 2): HTAB (14): PRINT Y$(J): NEXT J
2750  VTAB (12): INPUT "CHANGE THIS RECORD (Y/N)? ";A$
2760  IF A$ <  > "Y" THEN  HOME : GOTO 450
2762 LH% = 0
2770  VTAB (12): INVERSE
2772  PRINT "            INSTRUCTIONS                "
2774  PRINT "                                        "
2780  PRINT "HIT <CR> TO LEAVE AN ITEM UNCHANGED.    "
2790  PRINT "TO CHANGE, ENTER NEW VALUE <CR>.        "
2800  PRINT "TO DELETE THIS ENTRY, CHANGE THE CALL   "
2802  PRINT "     TO '*'.                            "
2804  NORMAL
2810  FOR J = 1 TO 8
2820  VTAB (J + 2): HTAB (14): INPUT "";G$: IF G$ <  > "" THEN Y$(J) = G$
2830  VTAB (J + 2): HTAB (14): PRINT Y$(J)
2840  NEXT J
2890  VTAB (19): INPUT "OK (Y/N)? ";A$:X1% = 4
2892  PRINT
2900  IF A$ <  > "Y" THEN  HOME : GOSUB 640: GOTO 450
2910  IF Y$(3) <  > Y$(13) THEN  GOSUB 3070: GOSUB 3100: REM  CALL CHANGED
2920  IF Y$(8) <  > Y$(18) THEN  GOSUB 3160: GOSUB 3190: IF BS = 1 THEN 450: REM
 SECTION CHANGED
2930  ON X1% GOTO 580,3210,3210,2940,2940,3220
2940  PRINT OP$;F3$;R3$;DV$
2950  PRINT WR$;F3$;",R";X2
2960  PRINT Y$(1);
2961  PRINT  SPC( 10 -  LEN (Y$(1)))Y$(2);
2962  PRINT  SPC( 6 -  LEN (Y$(2)))Y$(3);
2963  PRINT  SPC( 12 -  LEN (Y$(3)))Y$(4);
2964  PRINT  SPC( 7 -  LEN (Y$(4)))MD$;
2965  PRINT  SPC( 7 -  LEN (MD$))Y$(5);
2966  PRINT  SPC( 5 -  LEN (Y$(5)))Y$(6);
```

```
2967  PRINT " ";Y$(7);
2968  PRINT " ";Y$(8);
2969  PRINT  SPC( 5 -  LEN (Y$(8)))""
2970  PRINT CL$: IF HC$ <  > "Y" THEN 2990
2980  PRINT LP$: PRINT "  ** EDIT ON REC #";X2;": ";
2981  PRINT Y$(1);" ";Y$(2);" ";Y$(3);" ";Y$(4);" ";MD$;" ";Y$(5);Y$(6);Y$(7);Y$(8
)
2982  PRINT SCN$
2990  GOTO 450
3000  PRINT  TAB( 5)"EDITING RCD# ";X2: PRINT : PRINT  TAB( 5)"DATE:"
3010  PRINT  TAB( 5)"TIME:"
3020  PRINT  TAB( 5)"CALL:"
3030  PRINT  TAB( 5)"BAND:"
3031  PRINT  TAB( 5)"SERIAL#:"
3032  PRINT  TAB( 5)"PREC:"
3033  PRINT  TAB( 5)"CHECK:"
3040  PRINT  TAB( 5)"SECTION:"
3050  RETURN
3060  REM  CALL CHANGED
3070  IF Y$(3) = "*" THEN Y$(8) = "*": RETURN
3080 CS$ = Y$(3): GOSUB 3240: IF X1% < 4 THEN  RETURN : REM  CHECK IF NEW CALL IS
DUPE
3090  GOSUB 1040:TC% = TC% + 1: RETURN
3100  IF X1% < 4 THEN  RETURN
3110  IF Y$(13) = "*" THEN  RETURN
3120 CS$ = Y$(13)
3130  GOSUB 3240:C1%(H%) =  - 32001:C2%(H%) =  - 32000: REM  DELETE OLD CALL
3140 TC% = TC% - 1:X1% = 4: RETURN
3150  REM  EXCH CHANGED
3160  IF Y$(18) = "*" THEN  RETURN
3170 SK$ = Y$(18)
3180  GOTO 1340: REM  DELETE OLD SECTION
3190 BS = 0: IF Y$(8) = "*" THEN  RETURN
3200 SK$ = Y$(8): GOSUB 3620: IF BS = 1 THEN  RETURN
3205  GOSUB 1070: RETURN : REM  ADD NEW SECTION
3210  PRINT BL$;"INVALID CALL SIGN": GOTO 450
3220  PRINT BL$;"INVALID BAND": GOTO 450
3230  REM  DUPE CHECKER: X1%=1=DUPE: 2=QUERY: 3=INVALID CALL: 4,5=OK
3240 X1% = 4:E(5) = 0:E(6) = 0:K = 1: IF  LEN (CS$) > 10 THEN X1% = 3: RETURN
3250  IF  RIGHT$ (CS$,1) = "?" THEN CS$ =  LEFT$ (CS$, LEN (CS$) - 1):X1% = 2:CT$
= CS$
3260  IF  LEN (CS$) < 4 THEN X1% = 3: RETURN
3270  IF  MID$ (CS$,3,1) < "A" THEN 3300
3280  IF  MID$ (CS$,2,1) =  > "A" THEN X1% = 3: RETURN
3290 CS$ = " " + CS$:K = 2:E(1) = 0
3300  FOR J = K TO  LEN (CS$)
3310 W$ =  MID$ (CS$,J,1): IF W$ = "/" THEN 3360
3320 E(J) =  ASC (W$) - AA: IF E(J) > 26 THEN X1% = 3: RETURN
3330  IF W$ < "0" THEN X1% = 3: RETURN
3340  IF E(J) < 1 THEN E(J) = E(J) + A0: IF E(J) > 36 THEN X1% = 3: RETURN
3350  NEXT J
3360  IF J > 7 THEN X1% = 3: RETURN
3370  IF J - K < 4 THEN X1% = 3: RETURN
3380  FOR J = 4 TO 6: IF E(J) > 26 THEN X1% = 3: RETURN
3390  NEXT J
3400 E1% =  - 32000 + E(1) * 1369 + E(2) * 37 + E(3)
3410 E2% =  - 32000 + E(4) * 1369 + E(5) * 37 + E(6)
3420  IF X1% = 2 THEN  RETURN
3430 H% = E(4)
3440  REM  SEARCH DUPE STRING
3450  IF C1%(H%) =  - 32000 THEN X1% = 4: RETURN
3460  IF C2%(H%) <  > E2% THEN 3480
3470  IF C1%(H%) = E1% THEN X1% = 1: RETURN
3480  IF P%(H%) = 0 THEN X1% = 5: RETURN
3490 H% = P%(H%): GOTO 3460
3510  REM  SCORE
3515  PRINT : PRINT  SPC( 14)"SCO            RE SUMMARY": PRINT
3520  PRINT T1$
```

```
3530  GOSUB 3800
3550  PRINT T2$
3560  PRINT : PRINT  SPC( 9)"SCORE =";TC% * QP% * TS%
3570  PRINT : RETURN
3610  REM  SECTION CHECK
3620  FOR J = 1 TO NS: IF SK$ = S$(J) THEN 3650
3630  NEXT J
3640  PRINT BL$;"INVALID SECTION":BS = 1: RETURN
3650 SN = J:BS = 0: RETURN
3660  DATA  74,8,7,5,4,10,7,10,9,3,3,8
3661  REM  SECTION NAMES. MAX NAME LENGTH IS 4 CHARACTERS.
3670  DATA  "CO","IA" ,"KS","MN","MO","NE","ND","SD": REM  0'S
3680  DATA  "CT","EMA","ME","NH","RI","VT","WMA": REM  1'S
3690  DATA  "ENY","NLI","NNJ","SNJ","WNY": REM  2'S
3700  DATA  "EPA","DE","MDC","WPA": REM  3'S
3710  DATA  "AL","GA","KY","NC","NFL","SC","SFL","TN","VA","WIN": REM  4'S
3720  DATA  "AR","LA","MS","NM","NTX","OK","STX": REM  5'S
3730  DATA  "EB","LAX","ORG","SB","SCV","SDG","SF","SJV","SV","PAC": REM  6'S
3740  DATA  "AZ","ID","MT","NV","OR","UT","WA","WY","AK": REM  7'S
3750  DATA  "MI","OH","WV": REM  8'S
3760  DATA  "IL","IN","WI": REM  9'S
3770  DATA  "MAR","PQ","ON","MB","SK","AB","BC","VE8": REM  VE'S
3800 X1 = QP% * TC%
3810  PRINT "TOTALS  ";TC%;
3820  PRINT  SPC( 9 -  LEN ( STR$ (TC%)))QP%;
3830  PRINT  SPC( 7 -  LEN ( STR$ (QP%)))X1;
3840  PRINT  SPC( 10 -  LEN ( STR$ (X1)))TS%
3850  RETURN
3900  PRINT DTE$;
3910  PRINT  SPC( 10 -  LEN (DTE$))UTC$;
3920  PRINT  SPC( 6 -  LEN (UTC$))CT$;
3930  PRINT  SPC( 12 -  LEN (CT$))LN$;
3940  PRINT  SPC( 7 -  LEN (LN$))MD$;
3950  PRINT  SPC( 7 -  LEN (MD$))SN$;
3960  PRINT  SPC( 5 -  LEN (SN$))PC$;
3970  PRINT " ";CK$;
3980  PRINT " ";SK$;
3990  PRINT  SPC( 5 -  LEN (SK$))""
3999  RETURN
4000  IF RC$ <  > "Y" THEN 4020
4010  GOSUB 1182:U2$(ZA) = UTC$: GOTO 4030
4020  PRINT : INPUT "ENTER THE TIME IN UTC";U2$(ZA)
4030 UX =  VAL (U1$(ZA)):UY =  VAL (U2$(ZA))
4032 U7 =  INT (UX / 100):U4 = ((UX / 100) - U7) / .6:U7 = U7 + U4
4040 U5 =  INT (UY / 100):U6 = ((UY / 100) - U5) / .6:U5 = U5 + U6
4050  IF U5 < U7 THEN U5 = U5 + 24
4060 E7 = ET:ET = ET + U5 - U7:U3(ZA) = T: IF ET = 0 THEN 4090
4070  PRINT : PRINT "TIME USED SO FAR: "; INT (ET);" HRS, "; INT ((ET -  INT (ET))
 * 60 + .5);" MINS"
4080  PRINT : PRINT "OVERALL RATE:  "; INT ((T - 1) / ET + .5);" QSOS PER HOUR"
4090  RETURN
```

THE LOGPRINT PROGRAM

What it does

The LOGPRINT program reads the data files created by any of the three contest loggers and produces full logs after a contest is over. It will print the logs in the standard time-sequential form, or band by band.

Setup Instructions

LOGPRINT is set up for a standard 8½ × 11" page size, with top and bottom margins of 6 lines. This is the correct size for use with 9½ × 11" fan-folded tractor-feed forms. The paper length and top and bottom margins can be changed (see lines 80 and 90). If you're using wider paper, you can increase the tabbing, but it isn't necessary since the logs fit nicely within the 8½-inch width format that most contest sponsors expect.

Operating notes

Aside from turning on the printer and making sure the proper data files are on the disk in your first (or logged) disk drive, you don't have to do anything special to run LOGPRINT. You simply tell the program whether you want a time-sequential log or one with QSOs separated by band—that's all.

If you have modified the LOGGER programs to place data files on a drive other than the first one, you must make similar modifications in the LOGPRINT program. You can change the drive designation by changing ",D1" to ",D2" on the Apple version (line 72) or by inserting a drive name (e.g., "B:" or ":1") in the filenames at line 70 in the TRS-80 and CP/M version.

In some contests, the logs must show the exchange sent as well as the exchange received. Since the exchange sent does not vary in most contests, the LOGPRINT program does not print that information, nor must you enter it with each QSO when running the loggers. However, the record number, which can be used as a consecutive serial number, does appear on the screen and print out for each QSO. If the rules for a particular contest require that you show the exchange sent, write it in at the top of the first page. In some contests, the exchange sent includes a signal report. In practice, serious contest operators usually send 59 or 599 at all times, thus eliminating the need to record that information for each QSO. Thus, even in contests requiring signal reports, a single entry will usually take care of the exchange sent for the entire contest.

If you are printing logs and summary sheets for a multi-operator group effort in which more than one computer was used for logging, don't forget to tally up the grand total score and draw up an overall summary sheet. In most contests, the overall score cannot be determined merely by adding up the individual scores on various bands. Rather, the total score is usually determined by multiplying the cumulative multiplier total on all bands and the cumulative QSO point total on all bands.

Modification for CP/M

For CP/M systems running BASIC-80, the TRS-80 and CP/M listing should be changed as follows:

1. Delete the CLEAR statement in line 30;
2. Put your system's clear-screen command in place of "CLS" in line 130.

LOGPRINT (TRS-80 and CP/M)

```
10 REM LOG PRINTING PROGRAM (TRS-80 VERSION)
20 REM BY KC6A AND N6NB
30 CLEAR 250:BL$=CHR$(7):REM NEEDED; DON'T DELETE
60 FOR J=1 TO 48:PRINT:NEXT J
70 FA$="DUPELOGA":FC$="CONTSTLG":BL$=CHR$(7)
80 L1=66:REM PAGE LENGTH IN PRINT LINES
90 L2=6:REM TOP & BOTTOM MARGINS
100 CLS:INPUT "WHAT IS YOUR CALL ";MY$:MY$="LOG OF "+MY$
110 N1=0:N2=L1+L2+1
120 B$="":DIM U1$(15),U2$(15),U3(15)
130 CLS:PRINT:PRINT:PRINT TAB(27)"LOG SHEET"
140 PRINT TAB(24)"PRINTING PROGRAM":PRINT
150 PRINT TAB(16)"(C) 1983 BY HAYDEN BOOK CO.,INC."
160 PRINT:PRINT:PRINT
170 PRINT"POSITION THE PRINTER TO THE TOP OF A NEW PAGE."
180 INPUT"      HIT <CR> WHEN READY";G$
190 PRINT:PRINT:PRINT "DO YOU WANT ONE TIME-SEQUENTIAL LOG (T) OR A SEPARATE"
200 INPUT "LOG FOR EACH BAND (B) (ENTER T OR B)";TY$
210 IF TY$<>"T" THEN IF TY$<>"B" THEN 190
220 OPEN "I",1,FA$:OPEN "R",3,FC$,47
230 FIELD 3,8 AS D1$,4 AS TM$,10 AS KL$,5 AS B1$,5 AS M1$,15 AS E1$
240 INPUT#1,HC$:INPUT#1,T:INPUT#1,Y$:INPUT#1,Z$:INPUT#1,NL:INPUT#1,L%:INPUT#1,M1
250 DIM LN$(NL)
260 FOR J=1 TO NL:INPUT#1,LN$(J):NEXT J
262 INPUT#1,ZA:IF ZA=0 THEN 270
264 INPUT#1,ET:FOR J=1 TO ZA:INPUT#1,U1$(J):INPUT#1,U2$(J):INPUT#1,U3(J):NEXT J
270 G$=" "
280 LINE INPUT#1,G$:IF LEFT$(G$,1)<>BL$ THEN 280
300 LX=1:ZC=ZA:ZA=1
310 LT$="      ":LSET LT$=LN$(LX)
320 FOR REC=1 TO T-1
322 IF REC=U3(ZA) THEN GOSUB 680:ZA=ZA+1:N2=N2+1
330 GET 3,REC:REM GET LOG ENTRY
350 IF TY$="B" THEN IF LT$<>BAND$ THEN 370
360 IF LEFT$(KL$,1)<>"*" THEN GOSUB 520:REM PRINT IT IF NOT DELETED ENTRY
370 NEXT REC
372 IF ZC=>1 THEN LPRINT "   ***OFF THE AIR AT ";U2$(ZA);"***":N2=N2+1:ZA=1
380 IF TY$="B" THEN IF LX<NL THEN LX=LX+1:GOSUB 590:GOSUB 650:GOTO 320
390 CLOSE 3
400 GOSUB 590:REM NEW PAGE
410 IF EOF(1) THEN 470
420 LINE INPUT#1,G$
430 IF LEFT$(G$,1)=BL$ THEN GOSUB 590:GOTO 410
440 IF N2+L2>L1 THEN GOSUB 590:REM NEW PAGE
450 LPRINT G$:N2=N2+1
460 GOTO 410
470 IF ZC=>1 THEN LPRINT:LPRINT "TOTAL OPERATING TIME:  ";INT(ET);" HRS - ";INT(
    (ET-INT(ET))*60+.5);" MINS":N2=N2+1
472 GOSUB 640
480 PRINT:PRINT:PRINT TAB(27)"-DONE-":PRINT
490 PRINT"RUN DUPE PRINT PROGRAM TO GENERATE A DUPE SHEET."
500 END
510 REM PRINT A LINE
520 IF N2>L1-L2 THEN GOSUB 590:GOSUB 650:REM NEW PAGE
530 IF B1$=B$ THEN 550
540 LPRINT:N2=N2+1:B$=B1$:GOTO 520
550 LPRINT TAB(2)D1$;TAB(12)TM$;TAB(18)KL$;TAB(30)B1$;TAB(37)M1$;TAB(44)E1$;
560 LPRINT TAB(61)REC
570 B$=B1$:N2=N2+1:RETURN
580 REM PRINT TITLES - NEW PAGE
590 GOSUB 640:REM NEW PAGE
600 N1=N1+1
610 LPRINT TAB(6)Z$;TAB(36-LEN(MY$)/2)MY$;TAB(60)"PAGE ";N1
620 LPRINT TAB(36-LEN(Y$)/2)Y$:LPRINT:LPRINT
630 N2=N2+4:RETURN
640 FOR K=N2 TO L2+L1:LPRINT:NEXT K:N2=L2+1:RETURN
650 LPRINT TAB(3)"DATE";TAB(12)"UTC";TAB(19)"CALL";
```

```
660 LPRINT TAB(30)"BAND";TAB(37)"MODE";TAB(44)"EXCHANGE";TAB(61)"RCD#"
670 LPRINT:N2=N2+2:IF N1<>1 THEN RETURN
671 IF ZC<1 THEN RETURN
672 LPRINT "    ***ON THE AIR AT ";U1$(1);"***":N2=N2+1:RETURN
680 IF N2>L1-L2 THEN GOSUB 590:GOSUB 650
690 LPRINT "    ***OFF THE AIR AT ";U2$(ZA);" AND BACK ON AT ";U1$(ZA+1);"***"
700 RETURN
```

LOGPRINT (Apple)

```
10  REM  LOG PRINTING PROGRAM (APPLE)
20  REM  BY KC6A AND N6NB
40 BL$ =  CHR$ (7): REM  BELL
50 D$ =  CHR$ (4)
60 LP$ = D$ + "PR#1":SCN$ = D$ + "PR#0"
65 OP$ = D$ + "OPEN "
70 RD$ = D$ + "READ ":CL$ = D$ + "CLOSE "
72 DV$ = ",D1"
74 F1$ = "DUPELOGA"
76 F3$ = "CONTSTLG":R3$ = ",L58"
80 L1 = 66: REM  PAGE LENGTH IN PRINT LINES
90 L2 = 6: REM  TOP & BOTTOM MARGINS
100  HOME : INPUT "WHAT IS YOUR CALL? ";MY$:MY$ = "LOG OF " + MY$
110 N1 = 0:N2 = L1 + L2 + 1
120 B$ = "": DIM U1$(15),U2$(15),U3(15)
130  HOME : PRINT : INVERSE
135  PRINT "                                        "
140  PRINT "                LOG SHEET               "
145  PRINT "           PRINTING PROGRAM             "
148  PRINT "   (C) 1983 BY HAYDEN BOOK CO.,INC.     "
150  PRINT "                                        "
160  NORMAL : PRINT : PRINT
170  PRINT "POSITION THE PRINTER TO THE TOP OF"
180  INPUT "A NEW PAGE. HIT <CR> WHEN READY. ";G$
190  PRINT : PRINT : PRINT "DO YOU WANT ONE TIME-SEQUENTIAL LOG (T)"
192  INPUT "OR A SEPARATE LOG FOR EACH BAND (B)? ";TY$
200  IF TY$ <  > "T" THEN  IF TY$ <  > "B" THEN  PRINT "ENTER 'T' OR 'B'";BL$: GOT
O 190
210  PRINT OP$;F1$;DV$
220  PRINT RD$;F1$
230  INPUT HC$: INPUT T: INPUT Y$: INPUT Z$: INPUT NL: INPUT L%: INPUT M1
240  DIM LN$(NL)
250  FOR J = 1 TO NL: INPUT LN$(J): NEXT J
252  INPUT ZA: IF ZA = 0 THEN 260
254  INPUT ET: FOR J = 1 TO ZA: INPUT U1$(J): INPUT U2$(J): INPUT U3(J): NEXT J
260  PRINT CL$
300 LX = 1:ZC = ZA:ZA = 1
310 LT$ = LN$(LX)
312  IF  LEN (LT$) < 7 THEN  FOR J =  LEN (LT$) + 1 TO 7:LT$ = LT$ + " ": NEXT J
320  FOR REC = 1 TO T - 1
322  IF REC = U3(ZA) THEN  GOSUB 680:ZA = ZA + 1:N2 = N2 + 1
330  PRINT OP$;F3$;R3$;DV$
332  PRINT RD$;F3$;",R";REC
334  INPUT LE$
336  PRINT CL$
338 DT$ =  LEFT$ (LE$,10):UTC$ =  MID$ (LE$,11,6):KAL$ =  MID$ (LE$,17,12)
339 BD$ =  MID$ (LE$,29,7):MD$ =  MID$ (LE$,36,7):EXCH$ =  RIGHT$ (LE$,15)
340  IF TY$ = "B" THEN  IF LT$ <  > BD$ THEN 360
350  IF  LEFT$ (KAL$,1) <  > "*" THEN  GOSUB 510: REM  PRINT IT IF NOT DELETED ENT
RY
360  NEXT REC
362  IF ZC =  > 1 THEN  PRINT LP$: PRINT "    ***OFF THE AIR AT ";U2$(ZA);"***":N2
= N2 + 1:ZA = 1: PRINT SCN$
```

```
370  IF TY$ = "B" THEN  IF LX < NL THEN LX = LX + 1: GOSUB 580: GOSUB 640: GOTO 31
0
390  GOSUB 580: REM  NEW PAGE
392  PRINT OP$;F1$;DV$
393  PRINT RD$;F1$
395  INPUT G$: IF  LEFT$ (G$,1) <  > BL$ THEN 395
400  PRINT RD$;F1$
410  INPUT G$
420  IF  LEFT$ (G$,1) = BL$ THEN  GOSUB 580: GOTO 400: REM  NEW PAGE
422  IF G$ = "!" THEN 460: REM  EOF
430  IF N2 + L2 > L1 THEN  GOSUB 580: REM  NEW PAGE
440  PRINT LP$: PRINT G$: PRINT SCN$:N2 = N2 + 1
450  GOTO 400
460  IF ZC =  > 1 THEN  PRINT LP$: PRINT "TOTAL OPERATING TIME:  "; INT (ET);" HRS
 - "; INT ((ET -  INT (ET)) * 60 + .5);" MINS": PRINT SCN$:N2 = N2 + 1
462  PRINT CL$: GOSUB 630
470  PRINT : PRINT : PRINT  TAB( 16)"-DONE-": PRINT
480  PRINT  TAB( 9)"RUN DUPE PRINT PROGRAM"
485  PRINT  TAB( 7)"TO GENERATE A DUPE SHEET."
490  END
500  REM  PRINT A LINE
510  IF N2 > L1 - L2 THEN  GOSUB 580: GOSUB 640: REM  NEW PAGE
520  IF BD$ = B$ THEN 540
530  PRINT LP$: PRINT : PRINT SCN$:N2 = N2 + 1:B$ = BD$: GOTO 510
540  PRINT LP$: PRINT " ";DT$;UTC$;KAL$;BD$;MD$;EXCH$;
546  PRINT "  ";REC: PRINT SCN$
560 B$ = BD$:N2 = N2 + 1: RETURN
570  REM  PRINT TITLES - NEW PAGE
580  GOSUB 630: REM  NEW PAGE
590 N1 = N1 + 1
600  PRINT LP$: PRINT  SPC( 5)Z$; SPC( 30 -  LEN (Z$) -  LEN (MY$) / 2)MY$; SPC( 2
5 -  LEN (MY$))"PAGE ";N1
610  PRINT  SPC( 35 -  LEN (Y$) / 2)Y$: PRINT : PRINT
615  PRINT SCN$
620 N2 = N2 + 4: RETURN
630  IF N2 > L2 + L1 THEN 635
632  PRINT LP$
633  FOR K = N2 TO L2 + L1: PRINT : NEXT K
634  PRINT SCN$
635 N2 = L2 + 1: RETURN
640  PRINT LP$
650  PRINT "   DATE    UTC    CALL       BAND   MODE   EXCHANGE          RCD#"
660  PRINT : PRINT SCN$:N2 = N2 + 2: IF N1 <  > 1 THEN  RETURN
670  IF ZC < 1 THEN  RETURN
672  PRINT LP$: PRINT "   ***ON THE AIR AT ";U1$(1);"***":N2 = N2 + 1: PRINT SCN$:
 RETURN
680  IF N2 > L1 - L2 THEN  GOSUB 580: GOSUB 640
690  PRINT LP$: PRINT "   ***OFF THE AIR AT ";U2$(ZA);" AND BACK ON AT ";U1$(ZA +
1);"***": PRINT SCN$
700  RETURN
```

LOGPRINT (Commodore 64)

```
10 REM LOG PRINTING PROGRAM (C64 VERSION)
20 REM BY KC6A AND N6NB
30 REM COPYRIGHT 1983 BY HAYDEN BOOK CO.,INC.
32 OPEN 130,2,2,CHR$(136)+CHR$(1):REM RS232 PRINTER, 1200 BAUD/2 STOP BIT
34 OPEN 15,8,15
40 BL$=CHR$(7):REM BELL
50 CLS$=CHR$(147):REM PUT YOUR CLEAR-SCREEN COMMAND HERE
70 FA$="0:DUPELOGA":FC$="0:CONTSTLG"
80 L1=66:REM PAGE LENGTH IN PRINT LINES
90 L2=6:REM TOP & BOTTOM MARGINS
95 PRINT CLS$
110 N1=0:N2=L1+L2+1
120 B$="":DIM U1$(15),U2$(15),U3(15)
130 PRINT CLS$:PRINT:PRINT:PRINT TAB(15)"LOG SHEET"
```

```
140 PRINT TAB(12)"PRINTING PROGRAM":PRINT
150 INPUT "WHAT IS YOUR CALL";MY$:MY$="LOG OF "+MY$
160 PRINT:PRINT:PRINT
170 PRINT "POSITION THE PRINTER TO THE TOP OF A"
172 PRINT "NEW PAGE.  HIT <CR> WHEN READY."
180 G$="":INPUT G$
190 PRINT:PRINT:PRINT "DO YOU WANT ONE TIME-SEQUENTIAL LOG (T)"
192 PRINT "OR A SEPARATE LOG FOR EACH BAND (B)?"
200 INPUT "ENTER B OR T";TY$
210 IF TY$<>"T" THEN IF TY$<>"B" THEN GOTO 190
220 OPEN4,8,4,FA$+",S,R":GOSUB 6000
230 INPUT#4,HC$:INPUT#4,T
232 INPUT#4,Y$:INPUT#4,Z$:INPUT#4,NL
240 INPUT#4,L%:INPUT#4,M1
250 DIM LN$(NL)
260 FOR J=1 TO NL:INPUT#4,LN$(J):NEXT J
262 INPUT#4,ZA
263 IF ZA=0 THEN 310
264 INPUT#4,ET
266 FOR J=1 TO ZA:INPUT#4,U1$(J):INPUT#4,U2$(J):INPUT#4,U3(J):NEXT J
310 LX=1:ZC=ZA:ZA=1
315 OPEN3,8,3,FC$:GOSUB 6000
320 LT$=LN$(LX)
321 IF LEN(LT$)<5 THEN FOR J=LEN(LT$)+1 TO 5:LT$=LT$+"-":NEXT J
330 FOR REC=1 TO T-1
332 IF REC=U3(ZA) THEN GOSUB 680:ZA=ZA+1:N2=N2+1
340 GOSUB 7000
341 RP=1:GOSUB 7100:INPUT#3,D1$
342 RP=10:GOSUB 7100:INPUT#3,TM$
343 RP=15:GOSUB 7100:INPUT#3,KL$
344 RP=26:GOSUB 7100:INPUT#3,B1$
345 RP=32:GOSUB 7100:INPUT#3,M1$
346 RP=38:GOSUB 7100:INPUT#3,E1$
347 GOSUB 6000
350 IF TY$="B" AND B1$<>LEFT$(LT$,LEN(B1$)) THEN 370
360 IF LEFT$(KA$,1)<>"*" THEN GOSUB 520:REM PRINT IT IF NOT DELETED ENTRY
370 NEXT REC
372 IF ZC=>1 THEN PRINT#130,"    ***OFF THE AIR AT ";U2$(ZA);"****":N2=N2+1:ZA=1
380 IF TY$="B" AND LX<NL THEN LX=LX+1:GOSUB 590:GOSUB 650:GOTO 320
390 CLOSE 3
400 GOSUB 590:REM NEW PAGE
402 INPUT#4,G$:IF LEFT$(G$,1)<>BL$ THEN 402
403 IF G$="!" THEN 470
420 INPUT#4,G$
430 IF LEFT$(G$,1)=BL$ THEN GOSUB 590:GOTO 403
440 IF N2+L2>L1 THEN GOSUB 590:REM NEW PAGE
450 PRINT#130,G$:N2=N2+1
460 GOTO 403
470 IF ZC<1 THEN 474
471 PRINT#130,"":PRINT#130,"TOTAL OPERATING TIME: ";INT(ET);" HRS - ";
472 PRINT#130,"";INT((ET-INT(ET))*60+.5);" MINS":N2=N2+1
474 GOSUB 640
476 PRINT#130:FOR J=1 TO 5555:PRINT;:NEXT J
480 PRINT:PRINT:PRINT TAB(17)"-DONE-":PRINT
490 PRINT"RUN DUPE PRINT PROGRAM"
492 PRINT "TO GENERATE A DUPE SHEET."
502 CLOSE4:CLOSE130:CLOSE15:END
510 REM PRINT A LINE
520 IF N2>L1-L2 THEN GOSUB 590:GOSUB 650:REM NEW PAGE
530 IF B1$=B$ THEN 550
540 PRINT#130,"":N2=N2+1:B$=B1$:GOTO 520
550 PRINT#130,"  "D1$;SPC(10-LEN(D1$))TM$;SPC(6-LEN(TM$))KL$;
551 PRINT#130,"";SPC(12-LEN(KL$))B1$;SPC(7-LEN(B1$))M1$;SPC(7-LEN(M1$))E1$;
560 PRINT#130,"";SPC(17-LEN(E1$))REC
570 B$=B1$:N2=N2+1:RETURN
580 REM PRINT TITLES - NEW PAGE
590 GOSUB 640:REM NEW PAGE
600 N1=N1+1
610 PRINT#130,"       "Z$;SPC(30-(LEN(Z$)+LEN(MY$)/2))MY$;
611 PRINT#130,"";SPC(24-LEN(MY$))"PAGE ";N1
```

```
620 PRINT#130,"";SPC(36-LEN(Y$)/2)Y$:PRINT#130,"":PRINT#130,""
630 N2=N2+1:RETURN
640 FOR K=N2 TO L2+L1:PRINT#130,"":NEXT K:N2=L2+1:RETURN
650 PRINT#130,"   DATE      UTC    CALL       BAND    MODE    EXCHANGE          RCD#"
670 PRINT#130,"":N2=N2+2:IF N1<>1 THEN RETURN
671 IF ZC<1 THEN RETURN
672 PRINT#130,"   ***ON THE AIR AT ";U1$(1);"***":N2=N2+1:RETURN
680 IF N2>L1-L2 THEN GOSUB 590:GOSUB 650
690 PRINT#130,"   **OFF THE AIR AT ";U2$(ZA);" AND BACK ON AT ";U1$(ZA+1);"**"
700 RETURN
4701 F ZC<1 THEN 474
6000 INPUT#15,EA,EB$,EC,ED:IF EA<20 THEN RETURN:REM READ ERROR CHANNEL
6010 IF EA=50 THEN RETURN
6020 IF EA=63 THEN RETURN
6030 PRINT EA;EB$,EC;ED:STOP:RETURN
7000 RN=REC:RH=INT(RN/256):RL=RN-256*RH:RETURN
7100 PRINT#15,"P"CHR$(3)CHR$(RL)CHR$(RH)CHR$(RP)
7110 GOTO 6000
```

Conclusion

As experienced contest operators, we know how important rapid and accurate logging and dupechecking are to success in a contest. Considerable effort went into making the DUPECHECKER and the three contest loggers as versatile, fast, and powerful as possible. We couldn't write loggers especially for the unique rules of every contest, but these three logging programs will keep accurate logs for most of the popular contests. The DUPECHECKER will do the dupechecking for any contest, regardless of its scoring procedures.

When you log a contest with a computer and see how easy it is to keep track of dupes, multipliers, and your running score—and then discover a new freedom from post-contest drudgery—you'll probably wonder how we all managed before there were microcomputers. And if you use a computer for nothing more than retyping handwritten contest logs, you'll see that the ease of duping, the scoring accuracy and the neatness speak well for computerized logging. Contest logging and dupechecking are among the best applications for a computer in ham radio.

CHAPTER

9

PROGRAMS FOR ANTENNA DESIGN AND EVALUATION

Few subjects stir up more interest—or controversy—among radio amateurs than antennas. In the 40-plus years since the Yagi and cubical quad antenna designs became popular among amateurs, there have been thousands of over-the-air and in-person debates over the relative merits of the two antenna types for DX work on the high-frequency (HF) bands.

Not to be overshadowed by their HF brethren, VHF-UHF enthusiasts spend their time debating antennas, too. For example, some favor parabolic "dish" antennas for high-performance applications such as moonbounce, while others favor Yagis and other parasitic designs. Still others prefer collinear antennas.

Terms such as "radiation angle," "SWR," "pattern," "E-plane," "feedpoint impedance," "front-to-back ratio," and, above all, "decibels gain" are freely tossed about in these discussions. But heaven help the uninitiated newcomer who reveals his "shady" past by saying that two antennas are "co-phased" (a CB term that makes red-blooded amateurs bristle).

Much conflicting information has been published about various antennas, and even those who have set out to use computers to design the "optimum" antenna have often come up with results that proved less than optimum in field tests.

With some trepidation, we're gingerly stepping into this arena—at least in a small way. This chapter offers three antenna-oriented programs. They will not "prove" that any one kind of antenna is better than any other, but perhaps these programs will help you design antennas that will work as well as possible, and then evaluate the performance of your antennas once they're airborne.

Owners of computers without disk drives will be happy to know that these programs are less complex than the ones presented in the previous chapter: they will all work on almost any computer, with or without disk drives. Granted, even these programs will do fancier things if you have a disk drive and a printer. But they will do some of their tricks without any of these accoutrements.

The chapter includes an ANTENNA SCALER program, which will scale any beam antenna design to a new frequency, correcting for changes in the length-to-diameter ratio of the elements if you wish. And there's a program that will do a visual display of the patterns you get with phased (not "co-phased") vertical antennas in various configurations. It's called VERTICAL PATTERN PLOTTER, and it takes advantage of the high-resolution graphics features of an Apple II. If you don't have an Apple, it will still give you a video display (or hard-copy printout) of the radiation patterns, but with slightly less glamour. Finally, we offer an ANTENNA MATCHING EVALUATOR, a program that tells you the good (or bad) news about line losses, power at the antenna, and other details about your antenna system's performance.

The ANTENNA SCALER program is for hams, like co-author N6NB, who have a special interest in antennas. This photo shows one of N6NB's "moonbounce" expeditions. For this operation in the desert at the Utah-Nevada border, the antennas included a four-element 20-meter monobander on a 70-foot trailer-mounted tower (for HF liaison) and a two-meter "moonbounce" array of eight Quagi antennas.

THE ANTENNA SCALER

What it does

The ANTENNA SCALER will take the dimensions of any beam antenna and scale them to a new frequency. It offers three modes: SCALE, CREATE, and READ; the latter two are intended for computers with disk drives.

In the "SCALE" mode, you enter dimensions, specify a new frequency, and get a display (or printout) of dimensions for the new frequency.

If you have a disk drive, you can also use the other modes. The "CREATE" mode allows you to enter the dimensions of a proven antenna; the ANTENNA SCALER will create a sequential data file to save the dimensions on disk. Once you have created a data file, you can use the "READ" mode to scale antennas without having to enter the dimensions of the original antenna again and again. The program reads the data from disk; all you have to do is enter the information about the new antenna, such as the desired frequency and element diameter.

The program will correct the new dimensions for variations in length-to-diameter ratio, or it will do a straight mathematical scaling (ignoring the L/D ratio).

In using this program, don't forget that designing antennas by mathematical formula alone is a tricky business. If you design an antenna by formula (that is, by plugging in a frequency and computing dimensions), you're likely to get a set of theoretical dimensions that may or may not work in practice. There are many variables in the field that affect antenna performance. Any one of these variables can change the resonant frequency enough to throw off the computed dimensions.

If you base your computations on a field-tested design as this program does (instead of using a formula), you're more likely to get usable dimensions. But there are still random variables that can alter your results.

After you scale an antenna and build it, the first step is to plot its SWR at various frequencies, making sure the antenna is resonant where you expected it to be. If it isn't, all of the element lengths (not just the driven element length) should be adjusted. In fact, this program can be used to make those small adjustments, too. Enter the actual dimensions you used and the frequency where the antenna turned out to be resonant, and ask for dimensions for the frequency where you wanted the antenna resonant. The program will give you the new lengths.

Above all, don't expect miracles from an antenna scaling program. If you try to scale a design from one frequency to another many times less or greater, you may not get valid results. There are physical factors that you can't really scale. For instance, it isn't unusual for elements to be made of ⅛-inch-diameter solid rod at 1296 MHz, with the boom made of one-inch tubing. That would be the equivalent to using solid rod 12 inches thick for the elements of a 20-meter beam, with a boom eight feet in diameter! Even with the best corrections for length to diameter ratio, you'll get unpredictable results if you try to scale antennas over that kind of frequency range.

We have found the mathematical formulas in this program to be quite dependable for scaling antennas over a frequency range of about two to one. Beyond that, be prepared for some "tweaking" to get the desired results.

Setup instructions

No particular modifications are needed to run this program on the computers for which it was designed.

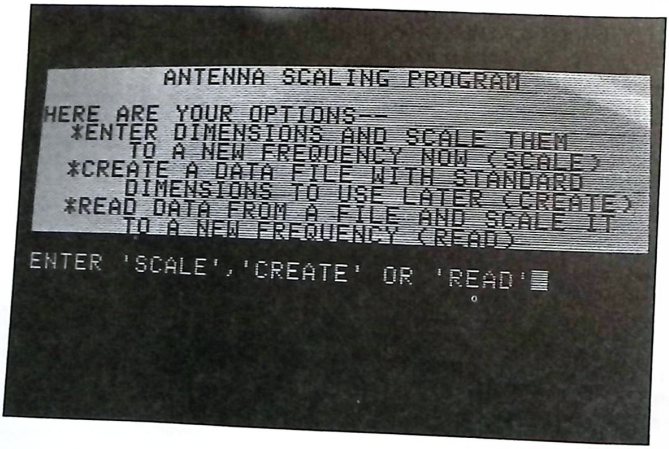
If you will be using it on a computer without disk drives, you can completely eliminate the subroutines that read and write data to sequential files on disk; these subroutines are clearly marked with REM statements. They begin at line 1590.

To shorten the program, the explanatory remarks about how the program works on various computers can also be eliminated. They are there primarily for demonstration purposes: at one time this program was used as an example of how the handling of hard-copy printouts and disk files differs on the Apple II, TRS-80, and CP/M systems running BASIC-80. The program will run on other computers having about 10K of free memory.

Operating notes

The program is menu-based and self explanatory.

The instructions say to enter the dimensions in inches because we have used the program primarily to scale VHF-UHF antennas. The dimensions could just as easily be entered in metric values—or in feet—*as long as all of the dimensions are in the same unit of measurement.* For instance, suppose you want to scale a proven 20-meter Yagi design to the 30-meter band. You could use feet in decimal form, entering the reflector length as 35.75 feet.


ANTENNA SCALING PROGRAM
HERE ARE YOUR OPTIONS--
*ENTER DIMENSIONS AND SCALE THEM
TO A NEW FREQUENCY NOW (SCALE)
*CREATE A DATA FILE WITH STANDARD
DIMENSIONS TO USE LATER (CREATE)
*READ DATA FROM A FILE AND SCALE IT
TO A NEW FREQUENCY (READ)
ENTER 'SCALE','CREATE' OR 'READ'

The ANTENNA SCALER program offers three options: antenna scaling in a direct-entry mode, the creation of a data file of standard dimensions, and antenna scaling based on a previously created data file.

Modifications for CP/M

For CP/M systems running BASIC-80, modify the TRS-80 and CP/M listing as follows:

1. Delete the CLEAR statement in line 35;
2. Put your video highlighting codes in line 60;
3. Replace the clear screen command ("CLS") with the proper command for your system (lines 50, 260, 660, and 910);
4. In the STRING$ commands (lines 300 and 575), replace the value 131 with 42 (131 is a special TRS-80 graphics character);
5. In the format for PRINT USING commands (lines 1060, 1070, 1180, and 1190) replace each percent symbol ("%") with a backslash ("\").

Sample printout of ANTENNA SCALER.

```
        ANTENNA SCALING FROM  144.2  TO  162.55 MHZ

                ANTENNA TYPE QUAGI144

ELEMENT   ORIG LGTH (IN)  NEW LGTH (IN)  NEW LGTH (CM)

REFL          86.50          76.73          194.9
DRIVEN        82.00          72.74          184.8
DIR 1         36.00          31.93           81.1
DIR 2         35.88          31.82           80.8
DIR 3         35.75          31.71           80.5
DIR 4         35.63          31.60           80.3
DIR 5         35.50          31.49           80.0
DIR 6         35.38          31.38           79.7

            INTER-ELEMENT SPACING DATA

ELEMENTS     ORIG SPACING (IN)      NEW SPACING (IN)

REF-DE             21.00                 18.63
DE-D1              15.75                 13.97
D 1-D2             33.00                 29.27
D 2-D3             17.50                 15.52
D 3-D4             26.10                 23.15
D 4-D5             26.10                 23.15
D 5-D6             26.10                 23.15

LENGTH-TO-DIAMETER RATIO CORRECTED

NOTE: MATHEMATICALLY DERIVED DIMENSIONS
MAY REQUIRE FINE TUNING IN THE FIELD.
```

Sample data file for ANTENNA SCALER.

```
6
36
35.875
35.75
35.625
35.5
35.375
82 , 86.5
21 , 15.75
33
17.5
26.1
26.1
26.1
144.2
.125
```

ANTENNA SCALER (TRS-80 and CP/M)

```
10 REM PARASITIC ANTENNA SCALING PROGRAM (TRS-80 VERSION)
20 REM FOR ANY QUAD, YAGI OR QUAGI OF TWO OR MORE ELEMENTS.
30 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
35 CLEAR 200
40 DIM A(25),E(25):B2=0:CM=2.54
50 CLS:REM CLEAR SCREEN
60 HI$="":LO$="":REM PUT YOUR VIDEO HIGHLIGHTING CODES HERE
120 PRINT HI$;"          ANTENNA SCALING PROGRAM          "
130 PRINT TAB(8)STRING$(23,131)
140 PRINT "HERE ARE YOUR OPTIONS--                    "
150 PRINT "  *ENTER DIMENSIONS AND SCALE THEM         "
160 PRINT "       TO A NEW FREQUENCY NOW (SCALE)      "
170 PRINT "  *CREATE A DATA FILE WITH STANDARD        "
180 PRINT "       DIMENSIONS TO USE LATER (CREATE)    "
190 PRINT "  *READ DATA FROM A FILE AND SCALE IT      "
200 PRINT "       TO A NEW FREQUENCY (READ)           "
220 PRINT LO$: INPUT "ENTER 'SCALE','CREATE' OR 'READ'";O$
230 IF O$="READ" THEN GOSUB 1820:GOTO 570
240 IF O$<>"CREATE" THEN IF O$<>"SCALE" THEN 140
250 REM DATA ENTRY SECTION
260 CLS:PRINT
280 PRINT HI$;"          ENTER THE DIMENSIONS          "
290 PRINT "        FOR THE ORIGINAL ANTENNA          ";LO$
300 PRINT TAB(8)STRING$(24,131):INPUT "HOW MANY DIRECTORS (IF NONE,'0')";N
310 IF N>25 THEN PRINT "TOO MANY. 25 MAXIMUM":GOTO 300
320 IF N<1 THEN 380
330 FOR J=1 TO N
340 PRINT:PRINT "ENTER THE LENGTH (IN INCHES)"
350 PRINT "   OF DIRECTOR #";J;
360 INPUT E(J)
370 NEXT J
380 PRINT:INPUT "HOW LONG IS THE DRIVEN ELEMENT";D:PRINT
390 INPUT "HOW LONG IS THE REFLECTOR";R:PRINT:PRINT:PRINT
400 PRINT
410 PRINT HI$;"      NOW ENTER THE ELEMENT SPACING      ":PRINT LO$
420 PRINT:INPUT "REFLECTOR TO DRIVEN ELEMENT";B1
430 IF N<1 THEN 500
440 PRINT:INPUT "DRIVEN ELEMENT TO DIRECTOR #1";B2
450 IF N<2 THEN 500
460 FOR J=1 TO (N-1)
470 PRINT:PRINT "DIRECTOR #";J;"TO DIRECTOR #";J+1
480 INPUT A(J)
490 NEXT J
500 PRINT:INPUT "WHAT IS THE ORIGINAL FREQUENCY (IN MHZ)";F1
510 PRINT:PRINT "WHAT IS THE ELEMENT DIAMETER (IN INCHES)"
520 PRINT "(USE TYPICAL DIAMETER AT CENTER IF IT VARIES)"
530 INPUT R1
540 IF O$="CREATE" THEN 1600
550 REM INFORMATION NEEDED TO SCALE TO A NEW FREQUENCY
570 PRINT:PRINT  HI$;"
575 PRINT TAB(8)STRING$(28,131)          NOW DESCRIBE THE NEW ANTENNA          ";LO$
580 PRINT:INPUT "ENTER THE NEW FREQUENCY (IN MHZ)";F2
590 SF=F1/F2:REM SCALING FACTOR
600 PRINT:PRINT "DO YOU WANT THE DIMENSIONS CORRECTED"
610 INPUT "FOR LENGTH-TO-DIAMETER RATIO (Y/N)";A$
620 IF A$<>"Y" THEN 650
630 PRINT :PRINT "WHAT IS THE NEW ELEMENT"
640 INPUT "DIAMETER (IN INCHES)";R2
650 IF SF<1.25 AND S>.8 THEN 830
660 CLS
670 PRINT:PRINT
690 PRINT HI$;"*****************************************"
700 PRINT "                                             "
710 PRINT "*****************WARNING                     "
720 PRINT LO$                                           "
730 PRINT "SCALING ANTENNAS MATHEMATICALLY IS A"
```

```
740 PRINT "TRICKY BUSINESS.  THERE ARE NUMEROUS"
750 PRINT "VARIABLES THAT AFFECT THE OUTCOME.  IF"
760 PRINT "THE CHANGE IN FREQUENCY IS SMALL, THEY"
770 PRINT "MAY BE IGNORED.  BUT WHEN A TWO-TO-ONE"
780 PRINT "OR GREATER CHANGE IS INVOLVED, SOME"
790 PRINT "EXPERIMENTATION WILL BE NECESSARY, EVEN"
800 PRINT "IF THE LENGTH-TO-DIAMETER RATIO IS TAKEN"
810 PRINT "INTO ACCOUNT IN THE SCALING."
830 PRINT:INPUT "DO YOU WANT A HARD COPY PRINTOUT? (Y/N)";LP$
840 X=R2/R1
850 L1=D/(.975-.6*R1/D)
860 L2=L1*SF
870 L3=L2*(.975-.6*R2/L2)
880 S=L3/D
890 IF A$<>"Y" THEN S=SF
900 REM DISPLAY AND PRINTOUT OF RESULTS
910 CLS
920 PRINT:PRINT
960 GOSUB 1020:REM FOR ALL COMPUTERS (VIDEO DISPLAY)
970 IF LP$<>"Y" THEN END
990 GOSUB 1340:REM PRINTING SUBROUTINE
1000 END
1010 REM VIDEO DISPLAY
1020 PRINT TAB(14)"ANTENNA SCALING FROM ";F1;" TO ";F2;"MHZ"
1030 PRINT
1040 IF O$="READ" THEN PRINT TAB(22)"ANTENNA TYPE ";FILE$:PRINT
1050 F1$="     ELEMENT    ORIG LGTH (IN)   NEW LGTH (IN)   NEW LGTH (CM)"
1060 F2$="     %     %       ###.##           ###.##           #,###.#"
1070 F3$="     %    %%        ###.##           ###.##           #,###.#"
1080 PRINT F1$
1090 PRINT USING F2$;"REFL",R,R*S,R*S*CM
1100 PRINT USING F2$;"DRIVEN",D,D*S,D*S*CM
1110 IF N<1 THEN 1145
1120 FOR J=1 TO N
1130 PRINT USING F3$;"DIR",J,E(J),E(J)*S,E(J)*S*CM
1140 NEXT J
1145 IF LP$<>"Y" THEN INPUT "ENTER <CR> TO CONTINUE";CR$
1150 PRINT:PRINT:PRINT
1160 PRINT TAB(19) "INTER-ELEMENT SPACING DATA":PRINT
1170 F4$="     ELEMENTS      ORIG SPACING (IN)      NEW SPACING (IN)"
1180 F5$="     %       %          ###.##                 ###.##"
1190 F6$="     %%%%%%             ###.##                 ###.##"
1200 PRINT F4$
1210 PRINT USING F5$;"REF-DE",B1,B1*SF
1220 IF N<1 THEN 1280
1230 PRINT USING F5$;"DE-D1",B2,B2*SF
1240 IF N<2 THEN 1280
1250 FOR J=1 TO N-1
1260 PRINT USING F6$;"D",J,"-D",J+1,A(J),A(J)*SF
1270 NEXT J
1280 PRINT:PRINT
1290 IF A$="Y" THEN PRINT "LENGTH-TO-DIAMETER RATIO CORRECTED"
1300 PRINT:PRINT "NOTE: MATHEMATICALLY DERIVED DIMENSIONS"
1310 PRINT "MAY REQUIRE FINE TUNING IN THE FIELD."
1320 RETURN
1330 REM HARD COPY PRINTOUT (EXCEPT APPLE)
1340 LPRINT TAB(14)"ANTENNA SCALING FROM ";F1;" TO ";F2;"MHZ"
1350 LPRINT:LPRINT
1360 IF O$="READ" THEN LPRINT TAB(22)"ANTENNA TYPE ";FILE$:LPRINT
1370 LPRINT F1$:LPRINT
1380 LPRINT USING F2$;"REFL",R,R*S,R*S*CM
1390 LPRINT USING F2$;"DRIVEN",D,D*S,D*S*CM
1400 IF N<1 THEN 1440
1410 FOR J=1 TO N
1420 LPRINT USING F3$;"DIR",J,E(J),E(J)*S,E(J)*S*CM
1430 NEXT J
1440 LPRINT:LPRINT:LPRINT
1450 LPRINT TAB(19) "INTER-ELEMENT SPACING DATA":LPRINT
1460 LPRINT F4$:LPRINT
```

```
1470 LPRINT USING F5$;"REF-DE",B1,B1*SF
1480 IF N<1 THEN 1540
1490 LPRINT USING F5$;"DE-D1",B2,B2*SF
1500 IF N<2 THEN 1540
1510 FOR J=1 TO N-1
1520 LPRINT USING F6$;"D",J,"-D",J+1,A(J),A(J)*SF
1530 NEXT J
1540 LPRINT:LPRINT
1550 IF A$="Y" THEN LPRINT "     LENGTH-TO-DIAMETER RATIO CORRECTED"
1560 LPRINT:LPRINT "     NOTE: MATHEMATICALLY DERIVED DIMENSIONS"
1570 LPRINT "     MAY REQUIRE FINE TUNING IN THE FIELD."
1580 RETURN
1590 REM CREATING A NEW DATA FILE FOR A SPECIFIC ANTENNA DESIGN
1600 PRINT "GIVE YOUR DATA FILE A NAME OF UP TO 8 CHARACTERS,"
1610 PRINT "PREFERABLY ONE THAT DESCRIBES THE ANTENNA."
1620 PRINT "THE FIRST CHARACTER MUST NOT BE A NUMERAL."
1630 INPUT "--NAME";FILE$
1640 IF LEN(FILE$)>8 THEN PRINT "TOO LONG":GOTO 1600
1650 OPEN "O",1,FILE$:REM TRS-80--'OPEN "O",1,FILE$'
1670 PRINT #1,N
1690 IF N<1 THEN 1710
1700 FOR J=1 TO N:PRINT #1,E(J):NEXT J
1710 PRINT #1,D;",";R
1720 PRINT #1,B1;",";B2
1730 IF N<2 THEN 1750
1740 FOR J=1 TO N-1:PRINT #1,A(J):NEXT J
1750 PRINT #1,F1:PRINT #1,R1
1760 CLOSE
1770 PRINT:PRINT "NEW FILE CREATED"
1780 END
1790 REM READING AN EXISTING DATA FILE--
1800 REM THE FILE HAS TO BE ON THE LOGGED DISK DRIVE AND THE
1810 REM NAME ENTERED MUST MATCH THE FILE NAME.
1820 PRINT:INPUT "WHAT IS THE NAME OF THE DATA FILE TO USE";FILE$
1830 OPEN "I",1,FILE$:REM TRS-80--'OPEN "I",1,FILE$'
1860 INPUT #1,N
1870 IF N<1 THEN 1890
1880 FOR J=1 TO N:INPUT #1,E(J):NEXT J
1890 INPUT #1,D,R:INPUT #1,B1,B2
1900 IF N<2 THEN 1920
1910 FOR J=1 TO N-1:INPUT #1,A(J):NEXT J
1920 INPUT #1,F1:INPUT #1,R1
1930 CLOSE
1940 RETURN
```

ANTENNA SCALER (Apple)

```
10  REM  PARASITIC ANTENNA SCALING PROGRAM (APPLE VERSION)
20  REM  FOR ANY QUAD, YAGI OR QUAGI OF TWO OR MORE ELEMENTS.
30  REM  BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
40  DIM A(25),E(25):B2 = 0:CM = 2.54
60  HOME : INVERSE
70 D$ =  CHR$ (4)
72 PR$ = D$ + "PR#1":SC$ = D$ + "PR#0"
120  PRINT "
130  PRINT "          ANTENNA SCALING PROGRAM          "
140  PRINT "HERE ARE YOUR OPTIONS--                    "
150  PRINT "  *ENTER DIMENSIONS AND SCALE THEM         "
160  PRINT "       TO A NEW FREQUENCY NOW (SCALE)      "
170  PRINT "  *CREATE A DATA FILE WITH STANDARD        "
180  PRINT "       DIMENSIONS TO USE LATER (CREATE)    "
```

```
190  PRINT "  *READ DATA FROM A FILE AND SCALE IT  "
200  PRINT "       TO A NEW FREQUENCY (READ)       "
210  NORMAL
220  PRINT : INPUT "ENTER 'SCALE','CREATE' OR 'READ'";O$
230  IF O$ = "READ" THEN  GOSUB 1820: GOTO 570
240  IF O$ <  > "CREATE" THEN  IF O$ <  > "SCALE" THEN 140
250  REM  DATA ENTRY SECTION
260  HOME : INVERSE
280  PRINT : PRINT "          ENTER THE DIMENSIONS          "
290  PRINT "       FOR THE ORIGINAL ANTENNA       "
300  NORMAL : PRINT : INPUT "HOW MANY DIRECTORS (IF NONE,'0')";N
310  IF N > 25 THEN  PRINT "TOO MANY. 25 MAXIMUM": GOTO 300
320  IF N < 1 THEN 380
330  FOR J = 1 TO N
340  PRINT : PRINT "ENTER THE LENGTH (IN INCHES)"
350  PRINT "   OF DIRECTOR #";J;
360  INPUT E(J)
370  NEXT J
380  PRINT : INPUT "HOW LONG IS THE DRIVEN ELEMENT?";D: PRINT
390  INPUT "HOW LONG IS THE REFLECTOR?";R: PRINT : PRINT : PRINT
400  INVERSE : PRINT
410  PRINT "     NOW ENTER THE ELEMENT SPACING      ": PRINT
420  NORMAL : PRINT : INPUT "REFLECTOR TO DRIVEN ELEMENT";B1
430  IF N < 1 THEN 500
440  PRINT : INPUT "DRIVEN ELEMENT TO DIRECTOR #1";B2
450  IF N < 2 THEN 500
460  FOR J = 1 TO (N - 1)
470  PRINT : PRINT "DIRECTOR #";J;" TO DIRECTOR #";J + 1
480  INPUT A(J)
490  NEXT J
500  PRINT : INPUT "ENTER THE ORIGINAL FREQUENCY (IN MHZ)";F1
510  PRINT : PRINT "ENTER THE ELEMENT DIAMETER (IN INCHES)"
520  PRINT "(USE DIAMETER AT CENTER IF IT VARIES)"
530  INPUT R1
540  IF O$ = "CREATE" THEN 1600
550  REM  INFORMATION NEEDED TO SCALE TO A NEW FREQUENCY
560  INVERSE
570  PRINT : PRINT "       NOW DESCRIBE THE NEW ANTENNA       "
580  NORMAL : PRINT : PRINT : INPUT "ENTER THE NEW FREQUENCY (IN MHZ)";F2
590 SF = F1 / F2: REM  SCALING FACTOR
600  PRINT : PRINT "DO YOU WANT THE DIMENSIONS CORRECTED"
610  INPUT "FOR LENGTH-TO-DIAMETER RATIO (Y/N)";A$
620  IF A$ <  > "Y" THEN 650
630  PRINT : PRINT "ENTER THE NEW ELEMENT"
640  INPUT "DIAMETER (IN INCHES)";R2
650  IF SF < 1.25 AND S > .8 THEN 830
660  HOME : INVERSE
670  PRINT : PRINT
690  PRINT "**************************************"
700  PRINT "                WARNING               "
710  PRINT "**************************************"
720  NORMAL : PRINT
730  PRINT "SCALING ANTENNAS MATHEMATICALLY IS A"
740  PRINT "TRICKY BUSINESS.  THERE ARE NUMEROUS"
750  PRINT "VARIABLES THAT AFFECT THE OUTCOME.  IF"
760  PRINT "THE CHANGE IN FREQUENCY IS SMALL, THEY"
770  PRINT "MAY BE IGNORED.  BUT WHEN A TWO-TO-ONE"
780  PRINT "OR GREATER CHANGE IS INVOLVED, SOME"
790  PRINT "EXPERIMENTATION WILL BE NECESSARY, EVEN"
800  PRINT "IF LENGTH-TO-DIAMETER RATIO IS TAKEN"
810  PRINT "INTO ACCOUNT IN THE SCALING."
820  PRINT : PRINT : PRINT
830  PRINT : INPUT "DO YOU WANT A HARD COPY PRINTOUT? (Y/N)";LP$
840 X = R2 / R1
850 L1 = D / (.975 - .6 * R1 / D)
860 L2 = L1 * SF
870 L3 = L2 * (.975 - .6 * R2 / L2)
880 S = L3 / D
890  IF A$ <  > "Y" THEN S = SF
900  REM  DISPLAY AND PRINTOUT OF RESULTS
```

```
910  DEF  FN A(X) =  INT (X * 100 + .5) / 100
920  DEF  FN B(X) =  INT (S * X * 100 + .5) / 100
930  DEF  FN C(X) =  INT (CM * S * X * 100 + .5) / 100
940  DEF  FN D(X) =  INT (SF * X * 100 + .5) / 100
960  HOME
970  IF LP$ = "Y" THEN  PRINT PR$
980  GOSUB 1020
1000  PRINT SC$: END
1010  REM  VIDEO DISPLAY (AND APPLE HARD COPY)
1020  PRINT "ANTENNA SCALING FROM ";F1;" TO ";F2;" MHZ"
1030  PRINT
1040  IF O$ = "READ" THEN  PRINT  TAB( 5)"ANTENNA TYPE ";FILE$: PRINT
1050  PRINT "ELE  ORIG.LGTH(IN)  NEW (IN)  NEW (CM)"
1060  PRINT
1090  PRINT "REFL"; SPC( 10 -  LEN ( STR$ ( FN A(R)))); FN A(R); SPC( 12 -  LEN (
STR$ ( FN B(R)))); FN B(R); SPC( 12 -  LEN ( STR$ ( FN C(R)))); FN C(R)
1100  PRINT "DE"; SPC( 12 -  LEN ( STR$ ( FN A(D)))); FN A(D); SPC( 12 -  LEN ( ST
R$ ( FN B(D)))); FN B(D); SPC( 12 -  LEN ( STR$ ( FN C(D)))); FN C(D)
1110  IF N < 1 THEN 1150
1120  FOR J = 1 TO N
1130  PRINT "DIR"; SPC( 11 -  LEN ( STR$ ( FN A(E(J))))); FN A(E(J)); SPC( 12 -  L
EN ( STR$ ( FN B(E(J))))); FN B(E(J)); SPC( 12 -  LEN ( STR$ ( FN C(E(J))))); FN C
(E(J))
1140  NEXT J
1145  IF LP$ <  > "Y" THEN  INPUT "ENTER <CR> TO CONTINUE";CR$
1150  PRINT : PRINT : PRINT
1160  PRINT  TAB( 7)"INTER-ELEMENT SPACING DATA": PRINT
1170  PRINT "ELE       ORIG SPACING (IN)      NEW  (IN)"
1210  PRINT "REF-DE"; SPC( 15 -  LEN ( STR$ ( FN A(B1)))); FN A(B1); SPC( 16 -  LE
N ( STR$ ( FN D(B1)))); FN D(B1)
1220  IF N < 1 THEN 1280
1230  PRINT "DE-D1"; SPC( 16 -  LEN ( STR$ ( FN A(B2)))); FN A(B2); SPC( 16 -  LEN
 ( STR$ ( FN D(B2)))); FN D(B2)
1240  IF N < 2 THEN 1280
1250  FOR J = 1 TO N - 1
1260  PRINT "D";J;"-D";J + 1; SPC( 16 -  LEN ( STR$ ( FN A(A(J))))); FN A(A(J)); S
PC( 16 -  LEN ( STR$ ( FN D(A(J))))); FN D(A(J))
1270  NEXT J
1280  PRINT : PRINT
1290  IF A$ = "Y" THEN  PRINT "LENGTH-TO-DIAMETER RATIO CORRECTED"
1300  PRINT : PRINT "NOTE: MATHEMATICALLY DERIVED DIMENSIONS"
1310  PRINT "MAY REQUIRE FINE TUNING IN THE FIELD."
1320  RETURN
1590  REM  CREATING A NEW DATA FILE FOR A SPECIFIC ANTENNA DESIGN
1600  PRINT "GIVE YOUR DATA FILE A NAME OF UP TO 8"
1610  PRINT "CHARACTERS, PREFERABLY ONE DESCRIPTIVE"
1620  PRINT "OF THE ANTENNA.  THE NAME MUST NOT"
1630  INPUT "BEGIN WITH A NUMERAL.  --THE NAME? ";FILE$
1640  IF  LEN (FILE$) > 8 THEN  PRINT "TOO LONG": GOTO 1600
1660  PRINT D$;"OPEN ";FILE$: PRINT D$;"DELETE ";FILE$
1665  PRINT D$;"OPEN ";FILE$: PRINT D$;"WRITE ";FILE$
1670  PRINT N
1690  IF N < 1 THEN 1710
1700  FOR J = 1 TO N: PRINT E(J): NEXT J
1710  PRINT D;",";R
1720  PRINT B1;",";B2
1730  IF N < 2 THEN 1750
1740  FOR J = 1 TO N - 1: PRINT A(J): NEXT J
1750  PRINT F1: PRINT R1
1760  PRINT D$;"CLOSE ";FILE$
1770  PRINT : PRINT "NEW FILE CREATED"
1780  END
1790  REM  READING AN EXISTING DATA FILE--
1800  REM  THE FILE HAS TO BE ON THE LOGGED DISK DRIVE AND THE
1810  REM  NAME ENTERED MUST MATCH THE FILE NAME.
1820  PRINT : INPUT "WHAT IS THE NAME OF THE DATA FILE TO USE";FILE$
1840  PRINT D$;"OPEN ";FILE$: PRINT D$;"READ ";FILE$
1860  INPUT N
1870  IF N < 1 THEN 1890
1880  FOR J = 1 TO N: INPUT E(J): NEXT J
```

```
1890  INPUT D,R: INPUT B1,B2
1900  IF N < 2 THEN 1920
1910  FOR J = 1 TO N - 1: INPUT A(J): NEXT J
1920  INPUT F1: INPUT R1
1930  PRINT D$;"CLOSE ";FILE$
1940  RETURN
```

ANTENNA SCALER (Commodore 64)

```
10 REM PARASITIC ANTENNA SCALING PROGRAM (C64)
20 REM FOR ANY QUAD, YAGI OR QUAGI OF TWO OR MORE ELEMENTS.
30 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
31 OPEN15,8,15
32 OPEN130,2,2,CHR$(136)+CHR$(1):REM RS232 1200-BAUD PRINTER
40 DIM A(25),E(25):B2=0:CM=2.54
50 CLS$=CHR$(147):REM CLEAR-SCREEN COMMAND
60 PRINT CLS$
70 INV$=CHR$(18):REM VIDEO-INVERSE COMMAND
80 NML$=CHR$(146):REM VIDEO-NORMAL COMMAND
110 PRINT " ";INV$;"                                    "
120 PRINT " ";INV$;"       ANTENNA SCALING PROGRAM      "
130 PRINT " ";INV$;"                                    ";:PRINT NML$
140 PRINT "HERE ARE YOUR OPTIONS--              "
150 PRINT "   ENTER DIMENSIONS AND SCALE THEM    "
160 PRINT "      TO A NEW FREQUENCY NOW (SCALE)  "
170 PRINT "   CREATE A DATA FILE WITH STANDARD   "
180 PRINT "      DIMENSIONS TO USE LATER (CREATE)"
190 PRINT "   READ DATA FROM A FILE AND SCALE IT "
200 PRINT "      TO A NEW FREQUENCY (READ)       "
220 PRINT:INPUT "ENTER 'SCALE','CREATE' OR 'READ'";O$
230 IF O$="READ" THEN GOSUB 1820:GOTO 570
240 IF O$<>"CREATE" THEN IF O$<>"SCALE" THEN 140
250 REM DATA ENTRY SECTION
260 PRINT CLS$:REM CLEAR SCREEN
270 PRINT:PRINT
280 PRINT "          ENTER THE DIMENSIONS          "
290 PRINT "        FOR THE ORIGINAL ANTENNA        "
300 PRINT:PRINT:PRINT "HOW MANY DIRECTORS (IF NONE,'0')":INPUT N
310 IF N>25 THEN PRINT "TOO MANY. 25 MAXIMUM":GOTO 300
320 IF N<1 THEN 380
330 FOR J=1 TO N
340 PRINT:PRINT "ENTER THE LENGTH (IN INCHES)"
350 PRINT "   OF DIRECTOR #";J;
360 INPUT E(J)
370 NEXT J
380 PRINT:PRINT "HOW LONG IS THE DRIVEN ELEMENT":INPUT D:PRINT
390 PRINT "HOW LONG IS THE REFLECTOR":INPUT R:PRINT:PRINT:PRINT
400 PRINT:PRINT
410 PRINT "      NOW ENTER THE ELEMENT SPACING      ":PRINT
420 PRINT:PRINT:INPUT "REFLECTOR TO DRIVEN ELEMENT";B1
430 IF N<1 THEN 500
440 PRINT:INPUT "DRIVEN ELEMENT TO DIRECTOR #1";B2
450 IF N<2 THEN 500
460 FOR J=1 TO (N-1)
470 PRINT:PRINT "DIRECTOR #";J;"TO DIRECTOR #";J+1
480 INPUT A(J)
490 NEXT J
500 PRINT:PRINT "WHAT IS THE ORIGINAL FREQUENCY (IN MHZ)":INPUT F1
510 PRINT:PRINT "WHAT IS THE ELEMENT DIAMETER (INCHES)"
520 PRINT "(USE TYPICAL DIAMETER IF IT VARIES)"
530 INPUT R1
540 IF O$="CREATE" THEN 1600
550 REM INFORMATION NEEDED TO SCALE TO A NEW FREQUENCY
560 PRINT
570 PRINT:PRINT  "        NOW DESCRIBE THE NEW ANTENNA        "
580 PRINT:INPUT "ENTER THE NEW FREQUENCY (IN MHZ)";F2
```

```
590 SF=F1/F2:REM SCALING FACTOR
600 PRINT:PRINT "DO YOU WANT THE DIMENSIONS CORRECTED"
610 PRINT "FOR LENGTH-TO-DIAMETER RATIO (Y/N)":INPUT A$
620 IF A$<>"Y" THEN 650
630 PRINT :PRINT "WHAT IS THE NEW ELEMENT DIAMETER"
640 INPUT "(IN INCHES)";R2
650 IF SF<1.25 AND S>.8 THEN 830
660 PRINT CLS$:REM CLEAR SCREEN
670 PRINT:PRINT
680 PRINT:PRINT
690 PRINT "*****************************************"
700 PRINT "                  WARNING                  "
710 PRINT "*****************************************"
720 PRINT:PRINT:PRINT NML$:REM VIDEO NORMAL
730 PRINT "SCALING ANTENNAS MATHEMATICALLY IS A"
740 PRINT "TRICKY BUSINESS.  THERE ARE NUMEROUS"
750 PRINT "VARIABLES THAT AFFECT THE OUTCOME.  IF"
760 PRINT "THE CHANGE IN FREQUENCY IS SMALL, THEY"
770 PRINT "MAY BE IGNORED.  BUT WHEN A TWO-TO-ONE"
780 PRINT "OR GREATER CHANGE IS INVOLVED, SOME"
790 PRINT "EXPERIMENTATION WILL BE NECESSARY, EVEN"
800 PRINT "IF LENGTH-TO-DIAMETER RATIO IS TAKEN"
810 PRINT "INTO ACCOUNT IN THE SCALING."
820 PRINT:PRINT:PRINT
830 PRINT:PRINT "DO YOU WANT A HARD COPY PRINTOUT? (Y/N)":INPUT LP$
840 X=R2/R1
850 L1=D/(.975-.6*R1/D)
860 L2=L1*SF
870 L3=L2*(.975-.6*R2/L2)
880 S=L3/D
890 IF A$<>"Y" THEN S=SF
900 REM DISPLAY AND PRINTOUT OF RESULTS
910 PRINT CLS$:REM CLEAR SCREEN
920 PRINT:PRINT
930 DEF FNA(X)=INT(X*100+.5)/100
932 DEF FNB(X)=INT(S*X*100+.5)/100
934 DEF FNC(X)=INT(CM*S*X*100+.5)/100
940 DEF FND(X)=INT(SF*X*100+.5)/100
960 GOSUB 1020:REM VIDEO DATA DISPLAY
970 IF LP$<>"Y" THEN 1780
990 GOSUB 1340:REM PRINTING SUBROUTINE
1000 GOTO 1780
1010 REM VIDEO DISPLAY
1020 PRINT "ANTENNA SCALING FROM ";F1;" TO ";F2;"MHZ"
1030 PRINT
1040 IF O$="READ" THEN PRINT TAB(5)"ANTENNA TYPE ";FILE$:PRINT
1050 F1$="ELE  ORIG L(IN)    NEW L(IN)    NEW L(CM)"
1080 PRINT F1$:PRINT
1090 PRINT "REFL";SPC(10-LEN(STR$(FNA(R))));FNA(R);
1092 PRINT SPC(12-LEN(STR$(FNB(R))));FNB(R);SPC(11-LEN(STR$(FNC(R))));FNC(R)
1100 PRINT "DE";SPC(12-LEN(STR$(FNA(D))));FNA(D);SPC(12-LEN(STR$(FNB(D))));
1102 PRINT FNB(D);SPC(11-LEN(STR$(FNC(D))));FNC(D)
1110 IF N<1 THEN 1150
1120 FOR J=1 TO N
1130 PRINT "DIR";SPC(11-LEN(STR$(FNA(E(J)))));FNA(E(J));
1132 PRINT SPC(12-LEN(STR$(FNB(E(J)))));FNB(E(J));
1134 PRINT SPC(11-LEN(STR$(FNC(E(J)))));FNC(E(J))
1140 NEXT J
1145 AA$="":INPUT "<CR> TO CONTINUE";AA$
1150 PRINT:PRINT
1160 PRINT TAB(7) "INTER-ELEMENT SPACING DATA":PRINT
1170 F4$="ELE     ORIG SPACING (IN)       NEW (IN)"
1200 PRINT F4$:PRINT
1210 PRINT "REF-DE";SPC(15-LEN(STR$(FNA(B1))));FNA(B1);
1212 PRINT SPC(16-LEN(STR$(FND(B1))));FND(B1)
1220 IF N<1 THEN 1280
1230 PRINT "DE-D1";SPC(16-LEN(STR$(FNA(B2))));FNA(B2);
1232 PRINT SPC(16-LEN(STR$(FND(B2))));FND(B2)
1240 IF N<2 THEN 1280
1250 FOR J=1 TO N-1
```

```
1260 PRINT "D";J;"-D";J+1;SPC(12-LEN(STR$(FNA(A(J)))));FNA(A(J));
1262 PRINT SPC(16-LEN(STR$(FND(A(J)))));FND(A(J))
1270 NEXT J
1280 PRINT:PRINT
1290 IF A$="Y" THEN PRINT "LENGTH-TO-DIAMETER RATIO CORRECTED"
1300 PRINT:PRINT "NOTE: MATHEMATICALLY DERIVED DIMENSIONS"
1310 PRINT "MAY REQUIRE FINE TUNING IN THE FIELD."
1320 RETURN
1330 REM HARD COPY PRINTOUT
1340 PRINT#130,"";SPC(14)"ANTENNA SCALING FROM ";F1;" TO ";F2;"MHZ"
1350 PRINT#130,"":PRINT#130,""
1360 IF O$="READ" THEN PRINT#130,"";SPC(22)"ANTENNA TYPE ";FILE$:PRINT#130,""
1370 PRINT#130,F1$:PRINT#130,""
1380 PRINT#130,"REFL";SPC(10-LEN(STR$(FNA(R))));FNA(R);
1382 PRINT#130,"";SPC(12-LEN(STR$(FNB(R))));FNB(R);
1384 PRINT#130,"";SPC(12-LEN(STR$(FNC(R))));FNC(R)
1390 PRINT#130,"DE";SPC(12-LEN(STR$(FNA(D))));FNA(D);
1392 PRINT#130,"";SPC(12-LEN(STR$(FNB(D))));FNB(D);
1394 PRINT#130,"";SPC(12-LEN(STR$(FNC(D))));FNC(D)
1400 IF N<1 THEN 1440
1410 FOR J=1 TO N
1420 PRINT#130,"DIR";SPC(11-LEN(STR$(FNA(E(J)))));FNA(E(J));
1422 PRINT#130,"";SPC(12-LEN(STR$(FNB(E(J)))));FNB(E(J));
1424 PRINT#130,"";SPC(12-LEN(STR$(FNC(E(J)))));FNC(E(J))
1430 NEXT J
1440 PRINT#130,"":PRINT#130,"":PRINT#130,""
1450 PRINT#130,"";SPC(19) "INTER-ELEMENT SPACING DATA":PRINT#130,""
1460 PRINT#130,F4$:PRINT#130,""
1470 PRINT#130,"REF-DE";SPC(15-LEN(STR$(FNA(B1))));FNA(B1);
1472 PRINT#130,"";SPC(16-LEN(STR$(FND(B1))));FND(B1)
1480 IF N<1 THEN 1540
1490 PRINT#130,"DE-D1";SPC(16-LEN(STR$(FNA(B2))));FNA(B2);
1492 PRINT#130,"";SPC(16-LEN(STR$(FND(B2))));FND(B2)
1500 IF N<2 THEN 1540
1510 FOR J=1 TO N-1
1520 PRINT#130,"D";J;"-D";J+1;SPC(12-LEN(STR$(FNA(A(J)))));FNA(A(J));
1522 PRINT#130,"";SPC(16-LEN(STR$(FND(A(J)))));FND(A(J))
1530 NEXT J
1540 PRINT#130,"":PRINT#130,""
1550 IF A$="Y" THEN PRINT#130,"     LENGTH-TO-DIAMETER RATIO CORRECTED"
1560 PRINT#130,:PRINT#130,"     NOTE: MATHEMATICALLY DERIVED DIMENSIONS"
1570 PRINT#130,"     MAY REQUIRE FINE TUNING IN THE FIELD."
1580 FOR J=1 TO 1999:PRINT;:NEXT J:RETURN
1590 REM CREATING A NEW DATA FILE FOR A SPECIFIC ANTENNA DESIGN
1600 PRINT:PRINT:PRINT "GIVE YOUR DATA FILE A NAME--8 CHARS MAX"
1610 PRINT "PREFERABLY ONE DESCRIBING THE ANTENNA."
1620 PRINT "THE FIRST CHARACTER MUST NOT BE A NUMERAL."
1630 INPUT "--NAME";FILE$
1640 IF LEN(FILE$)>8 THEN PRINT "TOO LONG":GOTO 1600
1644 PRINT#15,"S0:"+FILE$
1650 OPEN3,8,3,FILE$+",S,W"
1670 PRINT#3,N
1690 IF N<1 THEN 1710
1700 FOR J=1 TO N:PRINT#3,E(J):NEXT J
1710 PRINT#3,D;",";R
1720 PRINT#3,B1;",";B2
1730 IF N<2 THEN 1750
1740 FOR J=1 TO N-1:PRINT#3,A(J):NEXT J
1750 PRINT#3,F1:PRINT#3,R1
1760 CLOSE3
1770 PRINT:PRINT "NEW FILE CREATED"
1780 CLOSE130:CLOSE15:END
1790 REM READING AN EXISTING DATA FILE--
1800 REM THE FILE HAS TO BE ON THE LOGGED DISK DRIVE AND THE
1810 REM NAME ENTERED MUST MATCH THE FILE NAME.
1820 PRINT:PRINT "ENTER THE NAME OF THE DATA FILE TO USE":INPUT FILE$
1830 OPEN3,8,3,FILE$+",S,R"
1840 INPUT#15,EA,EB$,EC,ED:IF EA<20 THEN 1860
1850 IF EA=62 THEN PRINT EB$:CLOSE3:GOTO 1820
1860 INPUT#3,N
```

```
1870 IF N<1 THEN 1890
1880 FOR J=1 TO N:INPUT#3,E(J):NEXT J
1890 INPUT#3,D,R:INPUT#3,B1,B2
1900 IF N<2 THEN 1920
1910 FOR J=1 TO N-1:INPUT#3,A(J):NEXT J
1920 INPUT#3,F1:INPUT#3,R1
1930 CLOSE3
1940 RETURN
```

THE PHASED VERTICAL PATTERN DISPLAY

What it does

The VERTICAL PATTERN program will calculate the radiation patterns of many combinations of phased vertical antennas. It will display the calculated pattern on the screen and print a polar plot, using symbols. An accompanying photo shows the display as it appears on an Apple II.

This program also prints a table of the radiation pattern data for a phased vertical array at all radiation angles from the horizon to 90 degrees (i.e., straight up). The plot that displays on the screen and prints out in symbols is for the pattern on the horizon.

The program will calculate the results for anything from a single quarter-wave vertical to a complex array with many phased vertical elements. The length of each antenna, the distances between antennas, and the phase relationships are all considered in the computations.

AM broadcast engineers will recognize the formulas and the approach used by this program. Directional pattern computations such as this are vital in today's congested AM broadcast band. What is new in this program is the graphics capability and the printout simulating the polar plot of a phased vertical array.

Setup instructions

The Apple and TRS-80 – CP/M listings of this program differ considerably, since the Apple version of the program was written specifically for that computer's high-resolution graphics features. Therefore, the Apple version will run on nothing else; the other listing is much more generic. It uses a pattern of asterisks ("*") on the screen and the hard copy to simulate a polar plot.

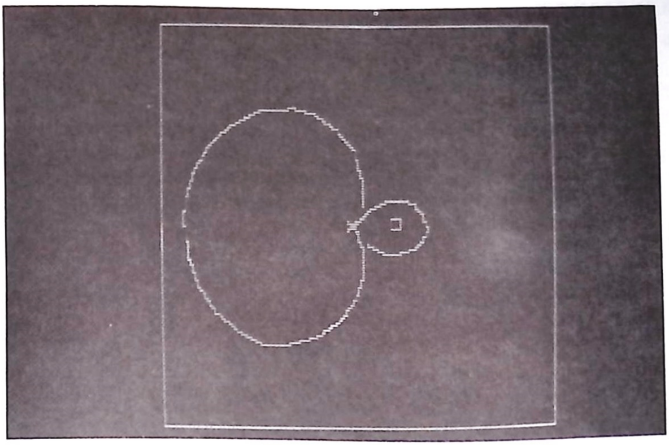
The program makes no use of disk files; it will run on a computer without disk drives, although use of a printer is desirable. It requires about 10K of free memory.

Operating notes

To run the VERTICAL PATTERN program, enter the parameters of the phased array for which you want a pattern plot and calculations.

First, indicate how many verticals there are in the array. If you type "1," you'll get the circular plot typical of a ground plane or quarter-wave vertical antenna. There is no limit to the number of verticals you could specify, other than the memory capacity of your computer.

For each additional antenna, enter its physical position in relation to the first antenna (called the "reference"). You must also input the height of each antenna, the spacing from the reference, the phase relationship, and the relative power to each antenna. The height, distance and phase



This is an example of a vertical antenna pattern plot, using the high-resolution graphics capabilities of an Apple II. This particular plot is for an array of two verticals fed 90 degrees out of phase.

relationship are all expressed in degrees, with 360 equaling one wavelength. The maximum permissible height is 359 degrees. The phase can be from −360 to +360. The input power to the reference antenna is assumed to be "1." If the power to the other antennas is the same, enter "1" as the power for each of the others as well. If the others only see half as much power, enter ".5" as their input power.

There is a delay in execution after you enter the number of antennas in the array if you ask for a hard copy; this is normal and not an indication of a problem in your computer.

The hard-copy printout shows the relative positions of the antennas with numbers inside the pattern. A "1" indicates the reference antenna (it will be at the center of the array), "2" is the second antenna for which you entered data, and so forth. On the Apple hi-res display, the reference is marked by an "X" in the center of the display. Other antennas are designated by squares.

It takes at least 24 lines—and often more—to display the full pattern. Almost all patterns will scroll off the 16-line screen of a TRS-80. However, the hard-copy printout is unaffected by this problem.

In interpreting the data shown in the table printed by this program, note that the relative field strengths at various elevation angles and bearings are just that: relative. The numbers produced by one run cannot be compared with those produced by other runs using other than exactly the same parameters. For example, the fact that the relative strength at 80 degrees elevation and 270 degrees azimuth may happen to be 2.5 with one set of parameters doesn't prove there's more signal in that direction than there may be when the level is shown as 1.5 during another run. The data table merely tells you how much of the signal goes in one direction as opposed to others with a given phased vertical configuration.

Sample printout from VERTICAL PATTERN

ANTENNA PATTERN FOR PHASED VERTICALS

ANT	AZ	SPACING	REL PH	REL PWR	HEIGHT
1	0	0	0	1	90
2	0	90	90	1	90
3	180	90	90	1	90

RELATIVE FIELD STRENGTH CALCULATIONS

	FIELD STRENGTH AT ELEVATION ANGLE							
AZ.	0	10	20	30	40	50	60	70	80
0	1	.98	.91	.82	.69	.56	.42	.28	.14
10	1	.98	.92	.82	.7	.56	.42	.28	.14
20	1.02	1	.93	.83	.71	.57	.43	.28	.14
30	1.08	1.06	.99	.88	.75	.61	.45	.3	.15
40	1.23	1.2	1.13	1.01	.86	.69	.51	.34	.17
50	1.46	1.43	1.33	1.19	1.01	.82	.61	.4	.2
60	1.73	1.69	1.58	1.41	1.2	.97	.72	.48	.24
70	1.99	1.94	1.82	1.62	1.38	1.11	.83	.55	.27
80	2.17	2.12	1.98	1.77	1.51	1.21	.91	.6	.3
90	2.24	2.19	2.04	1.83	1.55	1.25	.93	.62	.31
100	2.17	2.12	1.98	1.77	1.51	1.21	.91	.6	.3
110	1.99	1.94	1.82	1.62	1.38	1.11	.83	.55	.27
120	1.73	1.69	1.58	1.41	1.2	.97	.72	.48	.24
130	1.46	1.43	1.33	1.19	1.01	.82	.61	.4	.2
140	1.23	1.2	1.13	1.01	.86	.69	.51	.34	.17
150	1.08	1.06	.99	.88	.75	.61	.45	.3	.15
160	1.02	1	.93	.83	.71	.57	.43	.28	.14
170	1	.98	.92	.82	.7	.56	.42	.28	.14
180	1	.98	.91	.82	.69	.56	.42	.28	.14
190	1	.98	.92	.82	.7	.56	.42	.28	.14
200	1.02	1	.93	.83	.71	.57	.43	.28	.14
210	1.08	1.06	.99	.88	.75	.61	.45	.3	.15
220	1.23	1.2	1.13	1.01	.86	.69	.51	.34	.17
230	1.46	1.43	1.34	1.19	1.01	.82	.61	.4	.2
240	1.73	1.69	1.58	1.41	1.2	.97	.72	.48	.24
250	1.99	1.94	1.82	1.62	1.38	1.11	.83	.55	.27
260	2.17	2.12	1.98	1.77	1.51	1.21	.91	.6	.3
270	2.24	2.19	2.04	1.83	1.55	1.25	.93	.62	.31
280	2.17	2.12	1.98	1.77	1.51	1.21	.91	.6	.3
290	1.99	1.94	1.82	1.62	1.38	1.11	.83	.55	.27
300	1.73	1.69	1.58	1.41	1.2	.97	.72	.48	.24
310	1.46	1.43	1.33	1.19	1.01	.82	.61	.4	.2
320	1.23	1.2	1.12	1	.85	.69	.51	.34	.17
330	1.08	1.06	.99	.88	.75	.61	.45	.3	.15
340	1.02	1	.93	.83	.71	.57	.42	.28	.14
350	1	.98	.92	.82	.7	.56	.42	.28	.14

For someone planning an antenna system for 40, 80, or 160 meters, this program offers a very interesting and useful way to envision the results with various array configurations and phase relationships. If you have only limited real estate, it offers an opportunity to experiment with all of the combinations that might fit on your land without ever leaving your armchair.

Modifications for CP/M

For CP/M systems running BASIC-80, modify the TRS-80 and CP/M listing as follows:

1. Delete the CLEAR statement in line 45;
2. In line 38, replace CLS with your clear-screen command;
3. In line 770, delete the two left brackets ("[") and replace them with up-arrows ("^"). The TRS-80 uses a left bracket rather than the more typical up-arrow for exponentiation.

VERTICAL PATTERN (TRS-80 and CP/M)

```
10  REM  PHASED ANTENNA PATTERN DISPLAY PROGRAM (TRS-80)
20 REM --COMPUTES PHASED VERTICAL PATTERNS, DISPLAYS THEM
30 REM AND PRINTS A HARD COPY OF BOTH PATTERNS AND DATA.
36 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
38 CLS:REM CLEAR SCREEN
40  PRINT : PRINT  TAB(20)"PHASED VERTICAL ARRAY"
45 CLEAR 500
50 PRINT TAB(17)"DIRECTIONAL PATTERN DISPLAY": PRINT : PRINT
60 HC$="":INPUT "DO YOU WANT A HARD COPY? ";HC$
70  DEF FNM(X) = X - 2 *  INT (X / 2)
80 TM = 80:OF = .60
90 C = .0174553
100  IF HC$ <  > "Y" THEN TM = 1
140 PRINT:PRINT
150  PRINT "CHOOSE ONE ANTENNA AS THE REFERENCE."
160  PRINT "THEN SPECIFY THE POSITION AND PHASING"
170  PRINT "OF OTHER ANTENNAS IN RELATION TO IT."
175 PRINT:PRINT
180  INPUT "HOW MANY ANTENNAS IN THE ARRAY ";N
280  DIM G$(40,40)
290  FOR TX = 0 TO 39
300  FOR TY = 0 TO 39
310 G$(TX,TY) = " "
320  NEXT TY: NEXT TX
330 G$(20,20) = "1"
350  DIM AZ(N),SP(N),PH(N),EI(N),HT(N)
390  PRINT:PRINT "ENTER THE ELECTRICAL HEIGHT OF REFERENCE,"
392 PRINT "IN DEGREES  (NOTE: A 1/4-WAVE VERTICAL"
394 INPUT "IS 90 DEGREES).  # DEGREES";HT(1)
396 IF HT(1)>360 THEN PRINT "TOO MANY.  360 MAX.":GOTO 390
400 D=10:AZ(1) = 0:SP(1) = E:PH(1) = 0:EI(1) = 1
410 PRINT:PRINT
420 MM = 1
430  IF N = 1 THEN 520
440  FOR I = 2 TO N
450 PRINT:PRINT: PRINT  TAB(10)"DESCRIBE ANTENNA NO. ";I: PRINT
460 PRINT:PRINT "DIRECTION FROM REFERENCE IN DEGREES AZIMUTH?"
462 INPUT "(0=NORTH, 180=SOUTH, ETC.) "; AZ(I)
470 PRINT:PRINT "SPACING FROM REFERENCE IN DEGREES?"
472 INPUT "(1/4 WAVELENGTH=90)    " ;SP(I)
480 PRINT:INPUT "PHASE FROM REFERENCE IN DEGREES ";PH(I)
490 PRINT: PRINT "INPUT POWER RELATIVE TO REFERENCE ANT? "
492 INPUT "IF SAME AS REFERENCE, ENTER 1... ";EI(I)
500 MM = MM + EI(I)
510 PRINT: INPUT "ELECTRICAL HEIGHT, DEGREES";HT(I):  NEXT
520  IF HC$ = "Y" THEN  GOSUB 960
530 PRINT:PRINT
570  :TX = 80:TY = 80
600  IF HC$ <  > "Y" THEN 690
620  LPRINT  TAB(14)"RELATIVE FIELD STRENGTH CALCULATIONS"
630  FOR Q=1 TO 64:LPRINT "-";:NEXT Q
650  LPRINT
660  LPRINT  TAB(16)"FIELD STRENGTH AT ELEVATION ANGLE"
670  LPRINT "       0      10      20      30      40      50      60      70      80"
680  LPRINT "AZ.   ----    ----    ----    ----    ----    ----    ----    ----    ----"
690  FOR I = 0 TO 359 STEP D
700  IF HC$ <  > "Y" THEN 720
710  LPRINT I;TAB(6);
720  FOR T = 0 TO TM STEP 10
730 SX = 0:SY = 0
740  FOR M = 1 TO N
750  GOSUB 1100
760  NEXT M
770 RF =  SQR (SX [ 2 + SY [ 2)
780  IF T = 0 THEN  GOSUB 1200
790  GOSUB 1150
800  NEXT T
```

```
 810  IF HC$ = "Y" THEN  LPRINT
 820  NEXT I
 840 IF N=1 THEN 905
 850  FOR I = 2 TO N
 860 M = SP(I) / 360
 870 TX =  INT ( COS (AZ(I) * C) * M * 76 + 82):TY =  INT ( SIN (AZ(I) * C) * M *
       76 + 82)
 880 TX =  INT (TX / 4):TY =  INT (TY / 4)
 890  G$(TX,TY) =  STR$ (I)
 900  NEXT I
 905 GOSUB 2000
 910  IF HC$ = "Y" THEN  GOSUB 1320
 930  PRINT  CHR$ (7)
 950  END
 960  LPRINT
 970  LPRINT  TAB(10)"ANTENNA PATTERN FOR PHASED VERTICALS": LPRINT
 980  LPRINT "ANT  AZ  SPACING  REL PH  REL PWR  HEIGHT"
 990  LPRINT
1000  FOR K = 1 TO N
1010  LPRINT  TAB(2)K; TAB(6) AZ(K);
1020 LPRINT TAB(12) SP(K);
1030  LPRINT TAB(21) PH(K);
1040  LPRINT TAB(30) EI(K);
1050  LPRINT TAB(38) HT(K)
1060  NEXT K
1070  LPRINT
1090  RETURN
1100 A1 = AZ(M) * C:AA = I * C:P1 = PH(M) * C:V = HT(M) * C
1110 EL = T * C:NU =  COS (V *  SIN (EL)) -  COS (V):DN =  COS (EL) * (1 -  COS
      (V))
1120 F0 = NU / DN:EF = F0 * EI(M)
1130 BK = SP(M) * C *  COS (A1 - AA) + P1:X = EF *  COS (BK):Y = EF *  SIN (BK)
1140 SX = SX + X:SY = SY + Y: RETURN
1150 RF =  INT (100 * RF + .5) / 100
1160  IF HC$ <  > "Y" THEN  RETURN
1170 RS$ =  STR$ (RF)
1180 LPRINT RS$;STRING$((7-LEN(RS$)),32);
1190  RETURN
1200 TX = RF / MM * 19 *  COS (AA) + 20
1210 TY = RF / MM * 19 *  SIN (AA) + 20
1240  IF  INT (TX + .5) <  > 20 THEN 1290
1250  IF  INT (TY + .5) = 20 THEN  RETURN
1290 TX =  INT (TX):TY =  INT (TY)
1300 G$(TX,TY) = "*"
1310  RETURN
1320  LPRINT  CHR$(12): REM  FORM FEED
1330  LPRINT : LPRINT : LPRINT  TAB(34)"PATTERN PLOT"
1340  LPRINT : LPRINT  TAB(40)"0"
1350  FOR TX = 39 TO 0 STEP  - 1
1360  IF TX = 20 THEN 1390
1370  LPRINT "          ";
1380  GOTO 1400
1390  LPRINT "       270";
1400  FOR TY = 0 TO 39
1410  LPRINT G$(TX,TY);: IF  FNM(TY) = 1 THEN  LPRINT " ";
1420  NEXT TY
1430  IF TX <  > 20 THEN 1450
1440  LPRINT "90": GOTO 1460
1450  LPRINT
1460  NEXT TX
1470  LPRINT  TAB(39)"180"
1480  LPRINT : LPRINT : LPRINT
1500  RETURN
2000 PRINT:PRINT TAB(27)"PATTERN PLOT"
2010 PRINT:PRINT TAB(34)"0"
2020 FOR TX=39 TO 0 STEP -1
2030 IF TX=20 THEN 2060
2040 PRINT "   ";
2050 GOTO 2070
```

```
2060 PRINT "270";
2070 FOR TY=0 TO 39
2080 PRINT G$(TX,TY);:IF FNM(TY)=1 THEN PRINT " ";
2090 NEXT TY
2100 IF TX<>20 THEN 2120
2110 PRINT "90":GOTO 2130
2120 PRINT
2130 NEXT TX
2140 PRINT TAB(33)"180"
2150 RETURN
```

VERTICAL PATTERN (Apple)

```
10  REM   PHASED ANTENNA PATTERN DISPLAY PROGRAM (APPLE)
20  REM  --COMPUTES PHASED VERTICAL PATTERNS AND DISPLAYS
30  REM  THEM USING HI-RES GRAPHICS.  IT ALSO PRINTS HARD
32  REM  COPY OF PATTERNS AND DATA TABLES.
36  REM  BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
38  TEXT : HOME
40  PRINT : PRINT  TAB( 8)"PHASED VERTICAL ARRAY"
50  PRINT  TAB( 5)"DIRECTIONAL PATTERN DISPLAY": PRINT : PRINT
60  INPUT "DO YOU WANT A HARD COPY? ";HC$
70  DEF  FN M(X) = X - 2 *  INT (X / 2)
80 TM = 80:OF = 60
90 C = .0174553
100  IF HC$ <  > "Y" THEN TM = 1
110  PRINT : HOME
140  PRINT
150  PRINT "CHOOSE ONE ANTENNA AS THE REFERENCE."
160  PRINT "THEN SPECIFY THE POSITION AND PHASING"
170  PRINT "OF OTHER ANTENNAS IN RELATION TO IT."
180  PRINT : INPUT "HOW MANY ANTENNAS IN THE ARRAY? ";N
270  IF HC$ <  > "Y" THEN 340
280  DIM G$(40,40)
290  FOR TX = 0 TO 39
300  FOR TY = 0 TO 39
310 G$(TX,TY) = " "
320  NEXT TY: NEXT TX
330 G$(20,20) = "1"
340  HOME
350  DIM AZ(N),SP(N),PH(N),EI(N),HT(N)
360 D$ =  CHR$ (4)
370 LP$ = D$ + "PR#1"
380 SC$ = D$ + "PR#0"
390  PRINT : PRINT "ENTER ELECTRICAL HEIGHT OF REFERENCE"
392  PRINT "IN DEGREES  (NOTE: A 1/4-WAVE VERTICAL"
394  INPUT "IS 90 DEGREES).  # DEGREES? ";HT(1)
396  IF HT(1) > 360 THEN  PRINT "TOO MANY.  360 MAX.": GOTO 390
400 D = 10:AZ(1) = 0:SP(1) = E:PH(1) = 0:EI(1) = 1
420 MM = 1
430  IF N = 1 THEN 520
440  FOR I = 2 TO N
450  PRINT : PRINT : PRINT  TAB( 5)"DESCRIBE ANTENNA NO. ";I;":": PRINT
460  PRINT "AZIMUTH FROM REFERENCE, DEGREES? "
462  INPUT "(0=NORTH, 180=SOUTH, ETC.) ";AZ(I)
470  PRINT : PRINT "SPACING FROM REFERENCE, DEGREES? "
472  INPUT "(1/4 WAVELENGTH=90)    ";SP(I)
480  PRINT : INPUT "PHASE FROM REFERENCE, DEGREES? ";PH(I)
490  PRINT : PRINT "INPUT POWER RELATIVE TO REFERENCE ANT? "
492  INPUT "IF SAME AS REFERENCE, ENTER 1... ";EI(I)
500 MM = MM + EI(I)
510  PRINT : INPUT "ELECTRICAL HEIGHT, DEGREES? ";HT(I): NEXT
520  IF HC$ = "Y" THEN  GOSUB 960
570  HOME : HGR2 :TX = 80:TY = 80: GOSUB 1510
580  HCOLOR= 7
```

```
590  HPLOT OF,0 TO OF,159 TO 159 + OF,159 TO 159 + OF,0 TO OF,0
600  IF HC$ <  > "Y" THEN 690
610  PRINT LP$
620  PRINT  TAB( 14)"RELATIVE FIELD STRENGTH CALCULATIONS"
630  FOR Q = 1 TO 64: PRINT "-";
640  NEXT Q
650  PRINT
660  PRINT  TAB( 16)"FIELD STRENGTH AT ELEVATION ANGLE"
670  PRINT "      0      10      20      30      40      50      60      70      80"
680  PRINT "AZ.  ----    ----    ----    ----    ----    ----    ----    ----    ----"
690  FOR I = 0 TO 359 STEP D
700  IF HC$ <  > "Y" THEN 720
710  PRINT I;: HTAB (6)
720  FOR T = 0 TO TM STEP 10
730 SX = 0:SY = 0
740  FOR M = 1 TO N
750  GOSUB 1100
760  NEXT M
770 RF =  SQR (SX ^ 2 + SY ^ 2)
780  IF T = 0 THEN  GOSUB 1200
790  GOSUB 1150
800  NEXT T
810  IF HC$ = "Y" THEN  PRINT
820  NEXT I
830  HPLOT XO + OF,YO TO X1 + OF,Y1
840  IF N = 1 THEN 910
850  FOR I = 2 TO N
860 M = SP(I) / 360
870 TX =  INT ( COS (AZ(I) * C) * M * 76 + 82):TY =  INT ( SIN (AZ(I) * C) * M * 7
6 + 82)
880  GOSUB 1540:TX =  INT (TX / 4):TY =  INT (TY / 4)
890  IF HC$ = "Y" THEN G$(TX,TY) =  STR$ (I)
900  NEXT I
910  IF HC$ = "Y" THEN  GOSUB 1320
920  FOR J = 1 TO 30: PRINT : NEXT J
930  PRINT  CHR$ (7)
940  INPUT "  -DONE-   <CR> TO EXIT";A$
950  TEXT : HOME : END
960  PRINT LP$
970  PRINT : PRINT : PRINT  TAB( 10)"ANTENNA PATTERN FOR PHASED VERTICALS": PRINT
980  PRINT "ANT  AZ  SPACING  REL PH  REL PWR  HEIGHT"
990  PRINT
1000  FOR K = 1 TO N
1010  PRINT  TAB( 2)K;: HTAB (6): PRINT AZ(K);
1020  HTAB (12): PRINT SP(K);
1030  HTAB (21): PRINT PH(K);
1040  HTAB (30): PRINT EI(K);
1050  HTAB (38): PRINT HT(K)
1060  NEXT K
1070  PRINT
1080  PRINT SC$
1090  RETURN
1100 A1 = AZ(M) * C:AA = I * C:P1 = PH(M) * C:V = HT(M) * C
1110 EL = T * C:NU =  COS (V *  SIN (EL)) -  COS (V):DN =  COS (EL) * (1 -  COS (V
))
1120 F0 = NU / DN:EF = F0 * EI(M)
1130 BK = SP(M) * C *  COS (A1 - AA) + P1:X = EF *  COS (BK):Y = EF *  SIN (BK)
1140 SX = SX + X:SY = SY + Y: RETURN
1150 RF =  INT (100 * RF + .5) / 100
1160  IF HC$ <  > "Y" THEN  RETURN
1170 RS$ =  STR$ (RF)
1180  PRINT RS$; SPC( 7 -  LEN (RS$));
1190  RETURN
1200 TX = RF / MM * 19 *  COS (AA) + 20
1210 TY = RF / MM * 19 *  SIN (AA) + 20
1220 NX =  INT (TX * 4):NY =  INT (TY * 4)
1230  IF I = 0 THEN X1 = NX:Y1 = NY: GOTO 1270
1240  IF  INT (TX + .5) <  > 20 THEN 1260
1250  IF  INT (TY + .5) = 20 THEN XO = NX:YO = NY: RETURN
```

```
1260  HPLOT XO + OF,YO TO NX + OF,NY
1270 XO = NX:YO = NY
1280  IF HC$ <  > "Y" THEN  RETURN
1290 TX =  INT (TX):TY =  INT (TY)
1300 G$(TX,TY) = "*"
1310  RETURN
1320  PRINT LP$
1325  PRINT  CHR$ (12): REM   FORM FEED
1330  PRINT : PRINT : PRINT  TAB( 34)"PATTERN PLOT"
1340  PRINT : PRINT  TAB( 40)"0"
1350  FOR TX = 39 TO 0 STEP  - 1
1360  IF TX = 20 THEN 1390
1370  PRINT "           ";
1380  GOTO 1400
1390  PRINT "       270";
1400  FOR TY = 0 TO 39
1410  PRINT G$(TX,TY);: IF  FN M(TY) = 1 THEN  PRINT " ";
1420  NEXT TY
1430  IF TX <  > 20 THEN 1450
1440  PRINT "90": GOTO 1460
1450  PRINT
1460  NEXT TX
1470  PRINT  TAB( 39)"180"
1480  PRINT : PRINT : PRINT
1490  PRINT SC$
1500  RETURN
1510  HPLOT TX - 2 + OF,TY - 2 TO TX + 2 + OF,TY + 2
1520  HPLOT TX - 2 + OF,TX + 2 TO TX + 2 + OF,TY - 2
1530  RETURN
1540  HPLOT TX - 2 + OF,TY - 2 TO TX - 2 + OF,TY + 2 TO TX + 2 + OF,TY + 2 TO TX +
 2 + OF,TY - 2 TO TX - 2 + OF,TY - 2
1550  RETURN
```

THE ANTENNA MATCHING EVALUATOR

What it does

This short program, which will run on almost any computer with few changes, evaluates the performance of your antenna system, computing the consequences of SWR and feedline loss—and often producing surprising results.

The computations are based on forward power, either VSWR or reflected power, and the length and type of transmission line (the program includes loss data for several popular types of coaxial cable).

The program tells you the VSWR at the antenna (often quite different from the VSWR indicated at the transmitter), the true power into (and out of) the transmission line, and the amount of line loss.

Setup instructions

This program, which is reproduced here in only one standard version, will run on most computers with almost no modifications.

You should put your own system's clear-screen command in line 30 ("HOME" is for Apple; "CLS" works on TRS-80, "PRINT CHR$(147)" works for Commodore).

Because the TRS-80 uses a left bracket ("[") rather than the up-arrow ("^") for exponentiation, TRS-80 users must change the exponent symbol in lines 180, 505, 510, 550, 560, and 570. Those using the CP/M version of Microsoft BASIC-80 should note that they do *not* need to make this change.

If you want to add a hard-copy printout feature, you can do so by adding the proper commands for your system in lines 700–790.

You can add other cable types to the list contained in the DATA statements (see lines 2030–2050). The loss figures given are for 100 feet of cable at 100 MHz.

This program will not run as written on a Timex/Sinclair 1000 (ZX-81), because that computer does not understand the READ and DATA statements. The program will run on a Sinclair with 16K RAM if you omit the READ and DATA statements and enter the transmission line loss instead of having it computed.

This program will run on a VIC-20 without additional RAM if all optional program lines (i.e., lines 10, 20, 30, and 40) are deleted. Also, the lengthy input statements in lines 102, 130, 160, 260, 284, 310, and 400 should be modified for the VIC-20 as explained on page 69 in Chapter 5.

Some users may wish to perform a series of calculations without having to rerun the program each time. If you add a line that reads, "800 GOTO 100," the program will automatically rerun itself until you stop it with your computer's break or reset key.

Operating notes

The ANTENNA MATCHING EVALUATOR program needs some information from you to perform its computations. As we said earlier, you enter the power output and either the measured VSWR (at the transmitter) or the reflected power.

Next, enter the transmission line loss (if known) or just the type and length of line (and the frequency) so the program can compute the loss from a table of values for standard cable types. You have to enter a cable type the program recognizes—see the DATA statements in lines 2030–2050 for acceptable cable types.

When you have provided this information, the program will display the VSWR at the antenna, the true forward power, and the amount of power actually going to the antenna.

One possible source of difficulty is that the values you enter may not be mathematically possible. If, for instance, you enter a fairly high value for the VSWR at the transmitter and also a high value for feedline loss, the calculated VSWR at the antenna may exceed infinity, causing the program to print an error message and stop executing. What that tells you is simply this: even if the VSWR at the antenna were infinity, the indicated VSWR at the transmitter still wouldn't be as high as the value entered because the amount of feedline loss specified is sufficient to make an infinite VSWR at the antenna look like a lower value than that at the transmitter.

Modifications for CP/M

None required for CP/M, but see Setup Instructions.

ANTENNA MATCHING EVALUATOR (standard version)

```
10 REM ANTENNA MATCHING/TRANSMISSION LINE EVALUATION PROGRAM
20 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
32 PRINT:PRINT TAB(6)"ANTENNA MATCHING EVALUATOR"
40 REM CHANGE '^' TO '[' FOR TRS-80
50 PRINT:PRINT:PRINT
60 BL$=CHR$(7)
70 DEF FNL(X)=LOG(X)/LOG(10)
```

```
80 DEF FNA(X)=INT(X*100+.5)/100
90 PRINT:PRINT "NOTE:  ALL POWER LEVELS ARE IN WATTS"
100 PRINT:PRINT "WHAT IS YOUR FORWARD POWER READING"
102 INPUT "AT THE TRANSMITTER? ";FP
110 IF FP<=0 THEN PRINT BL$:GOTO 100
120 PRINT:PRINT "YOU'LL NEED TO ENTER THE REFLECTED"
122 PRINT "POWER OR THE VSWR.  WHICH DO YOU WISH--"
130 PRINT "REFLECTED POWER (R) OR VSWR (V)? ":INPUT A$
140 IF A$="R" THEN 190
150 IF A$<>"V" THEN PRINT BL$;"ENTER R OR V":GOTO 130
160 PRINT:PRINT "WHAT IS THE VSWR AT THE TRANSMITTER? ":INPUT VS
170 IF VS<1 THEN PRINT BL$:GOTO 160
180 RP=(((VS-1)/(VS+1))^2)*FP
185 GOTO 250
190 PRINT:PRINT "WHAT IS THE REFLECTED POWER AT THE"
192 INPUT "TRANSMITTER? ";RP
200 IF RP<0 THEN PRINT BL$;"ERROR":GOTO 190
210 IF RP>FP THEN PRINT BL$;"MUST BE LESS THEN FWD POWER":GOTO 190
220 V=SQR(RP/FP):VS=(1+V)/(1-V)
250 PRINT:PRINT "NOW YOU CAN ENTER EITHER THE TRANS-"
252 PRINT "MISSION LINE LOSS OR THE TYPE OF LINE"
254 PRINT "AND ITS LENGTH. "
260 PRINT:PRINT "WANT THE LINE LOSS CALCULATED? (Y/N) ":INPUT A$
270 IF A$="Y" THEN 310
280 PRINT:PRINT "WHAT IS THE TRANSMISSION LINE LOSS IN DB"
282 PRINT "NOTE--MUST BE MATHEMATICALLY POSSIBLE"
284 PRINT "WITH OTHER VALUES ENTERED.":INPUT LL
290 IF LL<0 THEN PRINT BL$;"ERROR": GOTO 280
300 GOTO 500
310 PRINT:INPUT "TRANSMISSION LINE= RG-";C$
315 REM FIND TRANSMISSION LINE LOSS
320 READ NM
330 FOR J = 1 TO NM
340 READ X$:READ LL
350 IF C$=X$ THEN 400
360 NEXT J
390 PRINT BL$;"INVALID LINE TYPE.  TRY AGAIN":GOTO 260
400 PRINT:INPUT "WHAT IS THE FREQUENCY IN MHZ? ";F
410 IF F<=0 THEN PRINT BL$:GOTO 400
420 PRINT:PRINT "HOW LONG IS THE TRANSMISSION LINE"
422 INPUT "IN FEET? ";LN
430 IF LN<0 THEN PRINT BL$;"ERROR":GOTO 420
440 LL=LL*SQR(F/100)*LN/100
500 REM CALC VSWR AT LOAD
505 IF VS=1 THEN VL=1:B=10^(LL/10):GOTO 530
510 A=(VS+1)/(VS-1):B=10^(LL/10)
520 VL=(A+B)/(A-B):REM VSWR AT LOAD
530 REM CALC LOSS IN CABLE
540 C=(VL-1)/(VL+1)
550 CL=10*FNL((B^2-C^2)/(B*(1-C^2))):REM CABLE LOSS
560 FL=FP/(10^(CL/10)):REM FWD POWER AT LOAD
570 RL=(((VL-1)/(VL+1))^2)*FL:REM REFL PWR AT LOAD
580 PR=FL-RL:REM POWER TO LOAD
600 PC=FP-RP-PR:REM POWER LOST IN CABLE
700 PRINT:PRINT
710 PRINT "VSWR AT TRANSMITTER=";TAB(30)FNA(VS)
720 PRINT "VSWR AT LOAD=";TAB(30)FNA(VL)
730 PRINT "FORWARD PWR AT XMTR=";TAB(30)FNA(FP)
740 PRINT "REVERSE PWR AT XMTR="; TAB(30)FNA(RP)
750 PRINT "TRUE POWER INTO XMSN LINE=";TAB(30)FNA(FP-RP)
760 PRINT "FORWARD POWER AT LOAD=";TAB(30)FNA(FL)
770 PRINT "REFLECTED PWR FROM LOAD=";TAB(30)FNA(RL)
780 PRINT "TRUE PWR INTO LOAD=";TAB(30)FNA(PR)
790 PRINT "LINE LOSS (INCL VSWR EFFECTS)=";TAB(30)FNA(CL);" DB"
1000 END
2000 REM TRANSMISSION LINE LOSS TABLE
2010 DATA 11:REM NUMBER OF ENTRIES
2020 REM TO EXPAND THIS TABLE, ENTER A TYPE OF LINE FOLLOWED BY
2022 REM ITS LOSS IN DB/100 FEET AT 100 MHZ
2030 DATA "8",2.3,"9",2.3,"11",2.3,"213",2.3,"214",2.3
2040 DATA "8FOAM",1.7,"17",.85,"1/2HARD",.85,"3/4HARD",.6
2050 DATA "58",5,"58FOAM",3.5
```

CHAPTER

10

TO THE MOON—AND BEYOND

Radio amateurs have always sought new challenges. Once everyone said you couldn't work DX except at night, and then only at frequencies below 200 meters. But hams pressed ever upward in frequency until they got to the dizzyingly high frequency of 14 MHz—20 meters. In 1924, two amateurs met the challenge and proved everyone wrong, working coast to coast at high noon.

When everyone said five meters was "line-of-sight" territory, hams in the 1930s proved that you could work amazing distances on that band by means of a newly discovered propagation mode called sporadic E-skip.

After World War II, everyone said you couldn't work long-haul DX on frequencies like 144 and 220 MHz. But in the late 1950s, two determined radio amateurs spanned the Pacific from California to Hawaii, first on two meters and then on 220. They barely missed making it on 432 MHz, and it was not until another generation of VHF DXers appeared on the scene in the 1970s that the challenge of spanning the Pacific on 432 was met.

And, of course, it was said for years after the U.S. Army succeeded in bouncing radar echoes off the moon that it couldn't be done with amateur power levels. But by now no fewer than 1,000 different amateur radio stations have completed two-way contacts over the difficult half-million-mile path from the earth to the moon and back.

Nevertheless, moonbounce remains one of the most difficult challenges in amateur radio, a real test of a ham's operating and station-building skills.

In this chapter, we offer three programs for those who have taken up the challenge of pointing their antennas toward the moon or beyond, or are considering it.

First, there is a program that will tell you whether your system has what it takes to get a signal to the moon and back. Or it will let you experiment with various power levels, antenna gains, and receiver noise figures until you find a combination that WILL do the job. The EME SYSTEM CALCULATOR considers all of the normal factors that enter into the EME (earth-moon-earth) equation and computes the probable results, in terms of how far above (or below) the noise level your returning echoes are likely to be.

For those who are ready to take on the challenge of moonbounce, we include a new version of one of the classic public-domain computer programs for amateur radio, the MOONTRACKER. In this version, you enter a date and tell the program how many days' worth of printouts you want, and it will perform all of the complex math required to prepare a chart that shows where in the sky the moon will be at various times on those dates.

The third program in this chapter is for those whose horizons extend even beyond the moon. A growing number of radio amateurs are pointing dishes toward the heavens, tracking celestial bodies that emit radio signals. The SKYLOCATOR program provides tracking data for this pursuit.

Although these programs are for those ready to take on some of the toughest and most exciting challenges in amateur radio, they will run on some of the least elaborate computer systems. None of these programs requires disk drives. And while a printer is quite desirable for the MOONTRACKER and SKYLOCATOR, they will run perfectly well without one if you're willing to forgo a hard copy of the resulting data. All that is required is about 10K of free memory. A Timex/Sinclair 1000 with a 16K memory module will do quite nicely.

THE EME SYSTEM CALCULATOR

What it does

The EME SYSTEM program uses formulas which compute system gain and path loss figures to determine what sort of signal you can expect to hear coming back from the moon if you try your hand at moonbounce with various equipment configurations.

You describe the system you own (or the system you'd like to have) in some detail. You list your power output, the frequency, your feedline losses, your antenna gain, and the noise figure and gain of your receiving preamps.

The program assumes average values for lunar path loss and sky temperature. The latter is set at 50 degrees Kelvin in line 100; you can change that value from the menu if you wish. It varies depending on the frequency and the portion of the sky that happens to be behind the moon at any given time.

Path loss at perigee (when the moon is closest to the earth) is about 1 dB. lower than the value calculated in the program, while the path loss at apogee (the point when the moon is the furthest from the earth) is about 1 dB. higher. In line 500, the constant 209.7 could be changed to approximate the path loss at perigee or apogee.

Once you have entered all of the parameters, the program will even tell you the approximate beamwidth of a typical antenna having the gain yours has, so you can determine how accurately you must point the antenna in order not to miss your target, the moon.

If you have or plan to have a dish antenna, the program will use a standard formula to compute the probable gain based on dish diameter and the frequency.

Setup instructions

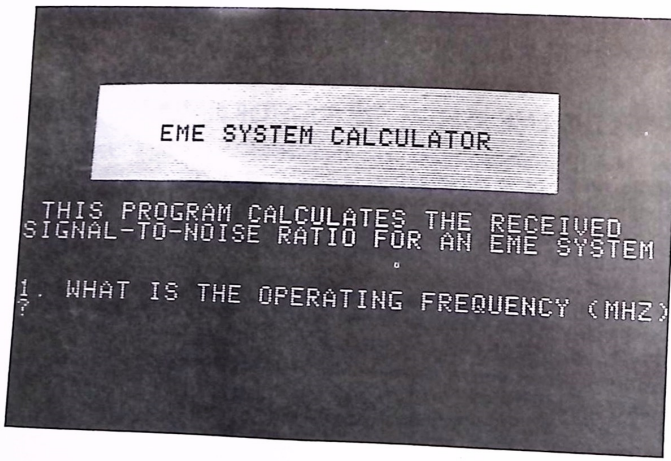
There are no major modifications likely to be needed to run this program on the computers for which it was written. In fact, the only major difference in the various versions of this program concerns the hard-copy printing functions. If you type the formulas accurately, it will run as listed.

Operating notes

In responding to the questions about your system, there are a number of options. Obviously, many of the values you enter will be estimates. Some may even be in the fantasy class. But if you want to see what it takes to have a big signal off the moon, go ahead—describe the station you'd like to have.

The program asks for the noise figure and gain of your first and second receiving preamps. If you use (or plan to use) just one, enter "0" for the gain and noise figure of the second preamp and also enter "0" for the line loss from the first preamp to the second.

If you specify a dish antenna, the formula contained in ITT's *Reference Data for Radio Engineers* is used to calculate the gain. For


EME SYSTEM CALCULATOR
THIS PROGRAM CALCULATES THE RECEIVED SIGNAL-TO-NOISE RATIO FOR AN EME SYSTEM
1. WHAT IS THE OPERATING FREQUENCY (MHZ)
?

EME (earth-moon-earth, or "moonbounce") communication represents one of the major challenges in amateur radio. It takes an outstanding station and sharp operating skills to do moonbounce work. The EME SYSTEM program won't size up your operating skills, but it will tell you whether your station is adequate for the task.

other antennas, you have to enter the gain in decibels *over isotropic* (dBi.). To determine the gain in dBi., add 2.17 to the gain in dB. over a dipole.

The program displays your probable antenna beamwidth, based on another formula from *Reference Data.* Since there's almost no signal to waste in most EME work, this is very important. For instance, if your 3 dB. beamwidth is 10 degrees, that means if you're even 5 degrees (not 10 degrees) off target you lose 3 dB. on both transmit and receive. Few EME systems are good enough to give away 6 dB. in roundtrip gain and still get good echoes. Thus, the program states the required aiming accuracy as half of the 1 dB. beamwidth, not the 3 dB. beamwidth.

The effect of receiver bandwidth on signal-to-noise ratio is also important, but sometimes misunderstood. If you specify a very narrow bandwidth, the S/N ratio may look great, but in practical terms the ringing of a high-selectivity filter may make copy difficult. Many EME operators actually find it easier to copy a signal with a wider bandwidth, even though the signal may theoretically be *below* the overall noise level under those conditions. Don't be intimidated if you get numbers showing that the received signal is 6 or 8 dB. below the noise level with a 2.7 kHz bandwidth. A good operator can often copy a signal under those conditions.

After you have run the program and seen the results, you can rerun it and change as few or as many of the values as you wish, without re-entering all the rest. A new menu will appear on the screen, allowing you to select the values you wish to revise.

Thus, you can try all sorts of combinations. You can see the result of replacing RG-8 with ⅞-inch hard line. Or you can try moonbounce with a really big dish antenna (like the 1000-footer at Arecibo) and see what can be done with 10 watts—or a kilowatt. Or you can enter your transmitting system parameters and someone else's receiving system parameters to see how strong your signal should be there.

Modifications for CP/M

For CP/M systems running BASIC-80, the TRS-80 and CP/M listing should be changed in these ways:

1. Delete the CLEAR statement in line 30;
2. Replace the left brackets ("[") in lines 80 and 90 with up-arrows ("^");
3. In line 160, change the value in the STRING$ command from 131 to 42;
4. Replace all CLS commands with your clear-screen command (see lines 130, 230, 250, 600, 1140 and 1700).

Sample printout from EME SYSTEM CALCULATOR

```
                EME SYSTEM CALCULATOR

             TRANSMITTER SPECIFICATIONS

FREQUENCY:                                  144   MHZ

TRANSMITTER OUTPUT POWER:                   700   WATTS
TRANSMITTER TO ANTENNA LINE LOSS:           1   DB
TRANSMIT ANTENNA:GAIN:                      24.1  DBI.
                 3DB BEAMWIDTH:             10.2  DEG.
                 APPROX POINTING ACCURACY:  1.7   DEG.
```

```
                RECEIVER SPECIFICATIONS

            LINE LOSS     GAIN     NF
PREAMP 1       .2  DB     17  DB    .5  DB
PREAMP 2       .5  DB     17  DB    .7  DB
CONVERTER      1   DB                2.7  DB
RECEIVER BANDWIDTH:                              .5  KHZ
RECEIVE ANTENNA:GAIN:                            24.1  DBI.
                3DB BEAMWIDTH:                   10.2  DEG.
                APPROX POINTING ACCURACY:        1.7  DEG.
RECEIVE SYSTEM TEMPERATURE:                      53.3  K

BASED ON A COLD SKY TEMPERATURE OF               50  DEGREES K
AND A MOON TEMPERATURE OF                        150  DEGREES K,
THE APPROXIMATE ANTENNA TEMPERATURE:             50.8  K
AND THE OVERALL SYSTEM TEMPERATURE:              104.2  K

EFFECTIVE RADIATED POWER LEVEL:                  81.5  DBM
MINUS PATH LOSS:                                -252.9  DB
PLUS RECEIVE ANTENNA GAIN:                       24.1  DBI
RECEIVED SIGNAL LEVEL:                          -147.3  DBM

SIGNAL TO NOISE RATIO:                           5.1  DB
```

EME SYSTEM CALCULATOR (TRS-80 and CP/M)

```
10  REM  EME SYSTEM CALCULATIONS (TRS-80 VERSION)
20  REM  BY KC6A AND N6NB.  (C) 1983 BY HAYDEN BOOK CO.,INC.
30 CLEAR 250
60  DEF  FN X(X) =  INT (10 * X) / 10
70  DEF  FN LG(X) =  LOG (X) /  LOG (10)
80  DEF  FN AL(X) = 10 [ (X / 10)
90  DEF  FN NT(X) = 293 * (10 [ (X / 10) - 1)
100 COLDSKY = 50: REM  COLD SKY TEMP IN K
110 MNTEMP = 150: REM  MOON TEMP IN DEG K
130 CLS:PRINT:PRINT
140 PRINT TAB(21)"EME SYSTEM CALCULATOR"
160 PRINT TAB(21)STRING$(21,131)
170  PRINT:PRINT "THIS PROGRAM CALCULATES THE RECEIVED"
180  PRINT "SIGNAL TO NOISE RATIO FOR AN EME SYSTEM"
190  PRINT
200  FOR ITM = 1 TO 12
210  ON ITM GOSUB 800,820,850,880,920,970,1000,1050,1080,1110,1140,1300
220  NEXT ITM
230  CLS:PRINT:INPUT "DO YOU WISH TO CHANGE ANYTHING (Y/N)";C$
240  IF C$ <  > "Y" THEN 410
250  CLS: PRINT : PRINT " 1.  FREQUENCY"
260  PRINT " 2.  TRANSMIT POWER"
270  PRINT " 3.  TRANSMIT LINE LOSS"
280  PRINT " 4.  RX ANT LINE LOSS"
290  PRINT " 5.  RX PREAMP #1"
300  PRINT " 6.  RX PREAMP #1 LINE LOSS"
310  PRINT " 7.  RX PREAMP #2"
320  PRINT " 8.  RX PREAMP #2 LINE LOSS"
330  PRINT " 9.  RX CONVERTER"
340  PRINT "10.  RX BANDWIDTH"
350  PRINT "11.  ANTENNA"
360  PRINT "12.  HARD COPY"
362  PRINT "13.  SKY TEMPERATURE"
370  PRINT : INPUT "WHICH ITEM DO YOU WISH TO CHANGE ";ITM
380  IF ITM > 13 THEN 230
390  ON ITM GOSUB 800,830,850,890,920,970,1000,1050,1080,1110,1150,1300,1700
400  GOTO 230
410  REM  CALCULATE TRANSMIT SIGNAL LEVEL
420  IF RD$ = "Y" THEN GX = 20 *  FN LG(FRQ) + 20 *  FN LG(RD) - 52.6
430  IF TD$ = "Y" THEN GT = 20 *  FN LG(FRQ) + 20 *  FN LG(TD) - 52.6
```

```
440 PT = 10 *  FN LG(PO) + 30 - TL + GT
450 LBTW2 = 10 *  FN LG(27000) - GT: REM  LOG TX BEAMWIDTH^2 (DEG)
460 BTW =  FN AL(LBTW2 / 2): REM  TX BW
470 BRW =  FN AL((10 *  FN LG(27000) - GX) / 2): REM  RX BEAMWIDTH
480 RS =  FN LG(.25) - (10 *  FN LG(27000) - GX): IF RS > 0 THEN RS = 0: REM  AR
    EA
490  REM  CALCULATE PATH ATTENUATION
500 ALPHA = 209.7 + 20 *  FN LG(FRQ): REM  PATH LOSS
510 RL = PT - ALPHA + GX: REM  RECEIVED SIGNAL LEVEL
520  REM  CALCULATE SYSTEM NOISE TEMP
530 M1 =  FN AL(L1):H1 =  FN AL(G1):O1 =  FN NT(N1)
540 M2 =  FN AL(L2):H2 =  FN AL(G2):O2 =  FN NT(N2)
550 M3 =  FN AL(L3):O3 =  FN NT(N3)
560 SKYTEMP = COLDSKY +  FN AL(RS) * (MNTEMP - COLDSKY)
570 SNT=293*(M1-1)+(O1*M1)+293*M1/H1*(M2-1)+O2*(M1*M2/H1)+293*(M3-1)*(M1*M2/(H1*
    H2))+O3*(M1*M2*M3/(H1*H2))
580 NOISE = 10 *  FN LG(SNT + SKYTEMP) + 10 *  FN LG(BW * 800) - 198.6
590 STN = RL - NOISE: REM  SIGNAL TO NOISE LEVEL IN DB
600  CLS
602  PRINT "FREQUENCY="; TAB( 27)FRQ;" MHZ"
610  PRINT "RCVR NOISE TEMPERATURE ="; TAB( 27)FN X(SNT);" K"
620  PRINT "SKY TEMPERATURE ="; TAB( 27) FN X(SKYTEMP);" K"
630  PRINT "SYSTEM NOISE TEMPERATURE ="; TAB( 27) FN X(SNT + SKYTEMP);" K"
640  PRINT "INPUT NOISE LEVEL ="; TAB( 27) FN X(NOISE);" DBM"
650  PRINT : PRINT "TRANSMIT ANTENNA GAIN = "; FN X(GT);" DBI"
660  PRINT  TAB( 8)"3DB BEAMWIDTH = "; FN X(BTW);" DEG"
670  PRINT "RECEIVE ANTENNA GAIN = "; FN X(GX);" DBI"
680  PRINT  TAB( 8)"3DB BEAMWIDTH = "; FN X(BRW);" DEG"
690  PRINT
700  PRINT : PRINT "TRANSMIT LEVEL ="; TAB( 27) FN X(PT);" DBM"
710  PRINT "PATH LOSS ="; TAB( 27) FN X(ALPHA);" DB"
720  PRINT "RECEIVED SIGNAL ="; TAB( 27) FN X(RL);" DBM"
730  PRINT : PRINT "SIGNAL TO NOISE RATIO = "; FN X(STN);" DB";
740  IF STN < 0 THEN  PRINT " (BELOW NOISE)";
750  PRINT
760  IF LP$ = "Y" THEN  GOSUB 1330
770  PRINT : INPUT "DO YOU WANT TO RE-RUN THE PROGRAM (Y/N) ";RR$
780  IF RR$ = "Y" THEN 230
790  END
800  PRINT : INPUT "1. WHAT IS THE OPERATING FREQUENCY (MHZ) ";FRQ
810  RETURN
820 PRINT:PRINT: PRINT  TAB( 12)"TRANSMITTER SPECIFICATIONS": PRINT
830  PRINT : INPUT "2. WHAT IS THE TRANSMITTER OUTPUT POWER (WATTS) ";PO
840  RETURN
850  PRINT : PRINT "3. WHAT IS THE TRANSMITTER TO ANTENNA"
860  INPUT "    LINE LOSS IN DB ";TL
870  RETURN
880 PRINT:PRINT: PRINT  TAB( 12)"RECEIVER SPECIFICATIONS": PRINT
890  PRINT : PRINT "4. WHAT IS THE LINE LOSS FROM THE"
900  INPUT "     ANTENNA TO THE 1ST PREAMP IN DB ";L1
910  RETURN
920  PRINT : PRINT "5. WHAT IS THE GAIN OF THE 1ST"
930  INPUT "    PREAMP IN DB ";G1
940  PRINT : PRINT "   WHAT IS THE NOISE FIGURE OF THE"
950  INPUT "    1ST PREAMP IN DB ";N1
960  RETURN
970  PRINT : PRINT "6. WHAT IS THE LINE LOSS FROM THE 1ST"
980  INPUT "    PREAMP TO THE 2ND PREAMP IN DB ";L2
990  RETURN
1000  PRINT : PRINT "7. WHAT IS THE GAIN OF THE 2ND"
1010  INPUT "    PREAMP IN DB ";G2
1020  PRINT : PRINT "   WHAT IS THE NOISE FIGURE OF THE"
1030  INPUT "    2ND PREAMP IN DB ";N2
1040  RETURN
1050  PRINT : PRINT "8. WHAT IS THE LINE LOSS FROM THE 2ND"
1060  INPUT "    PREAMP TO THE CONVERTER IN DB ";L3
1070  RETURN
1080  PRINT : PRINT "9. WHAT IS THE NOISE FIGURE OF THE"
1090  INPUT "    RECEIVE CONVERTER IN DB ";N3
```

```
1100  RETURN
1110  PRINT : PRINT "10. WHAT IS THE RECEIVE BANDWIDTH"
1120  INPUT "     IN KHZ ";BW
1130  RETURN
1140 CLS
1145 PRINT:PRINT TAB(12)"ANTENNA PARAMETERS":PRINT
1150  INPUT "11. IS THE RECEIVE ANTENNA A DISH (Y/N) ";RD$
1160  PRINT
1170  IF RD$ = "Y" THEN 1200
1180  PRINT "     WHAT IS THE RECEIVE ANTENNA GAIN"
1190  INPUT "     OVER AN ISOTROPIC SOURCE IN DB ";GX: GOTO 1210
1200  INPUT "WHAT IS THE DIAMETER IN FEET ";RD
1210  PRINT : INPUT "IS THE TRANSMIT ANTENNA THE SAME (Y/N) ";TT$
1220  IF TT$ = "Y" THEN GT = GX:TD$ = RD$:TD = RD: GOTO 1290
1230  PRINT : INPUT "     IS THE TRANSMIT ANTENNA A DISH ";TD$
1240  PRINT
1250  IF TD$ = "Y" THEN 1280
1260  PRINT "     WHAT IS THE TRANSMIT ANTENNA GAIN"
1270  INPUT "     OVER AN ISOTROPIC SOURCE IN DB ";GT: GOTO 1290
1280  INPUT "     WHAT IS THE DIAMETER IN FEET ";TD
1290  RETURN
1300  PRINT : INPUT "DO YOU WANT HARD COPY (Y/N) ";LP$
1302  IF LP$ = "Y" THEN  GOSUB 1800: REM  INIT PRINTER, IF NEEDED
1310  RETURN
1330  LPRINT  TAB( 20)"EME SYSTEM CALCULATOR"
1340  LPRINT : LPRINT  TAB( 17)"TRANSMITTER SPECIFICATIONS"
1341  LPRINT
1342  LPRINT "FREQUENCY: ";TAB(45)FRQ;" MHZ"
1350  LPRINT : LPRINT "TRANSMITTER OUTPUT POWER: ";TAB(45)PO;" WATTS"
1360  LPRINT "TRANSMITTER TO ANTENNA LINE LOSS: ";TAB(45)TL;" DB"
1370  LPRINT "TRANSMIT ANTENNA: ";
1380  IF TD$ = "Y" THEN  LPRINT TD;" FOOT DISH. ";
1390  LPRINT "GAIN: "; TAB(45)FN X(GT);" DBI."
1392  LPRINT  TAB( 18)"3DB BEAMWIDTH: "; TAB(45)FN X(BTW);" DEG."
1394  LPRINT  TAB( 18)"APPROX POINTING ACCURACY: "; TAB(45)FN X(BTW / 6);" DEG."
1400  LPRINT : LPRINT  TAB( 19)"RECEIVER SPECIFICATIONS"
1410  LPRINT : LPRINT  TAB( 12)"LINE LOSS     GAIN     NF"
1420  LPRINT "PREAMP 1     ";L1;" DB";"     ";G1;" DB";SPC(3)N1;" DB"
1430  LPRINT "PREAMP 2     ";L2;" DB";"     ";G2;" DB";SPC(3)N2;" DB"
1440  LPRINT "CONVERTER    ";L3;" DB";"                ";N3;" DB"
1442  LPRINT "RECEIVER BANDWIDTH:"; TAB(45)BW;" KHZ"
1450  LPRINT "RECEIVE ANTENNA: ";
1460  IF RD$ = "Y" THEN  LPRINT RD;" FOOT DISH--";
1470  LPRINT "GAIN: "; TAB(45)FN X(GX);" DBI."
1472  LPRINT  TAB( 18)"3DB BEAMWIDTH: "; TAB(45)FN X(BRW);" DEG."
1474  LPRINT  TAB( 18)"APPROX POINTING ACCURACY: "; TAB(45)FN X(BRW / 6);" DEG."
1480  LPRINT "RECEIVE SYSTEM TEMPERATURE:"; TAB(45)  FN X(SNT);" K"
1490  LPRINT
1500  LPRINT "BASED ON A COLD SKY TEMPERATURE OF ";TAB(45)COLDSKY;" DEGREES K"
1510  LPRINT "AND A MOON TEMPERATURE OF ";TAB(45)MNTEMP;" DEGREES K,"
1520  LPRINT "THE APPROXIMATE ANTENNA TEMPERATURE: ";TAB(45)FN X(SKYTEMP);" K"
1530  LPRINT "AND THE OVERALL SYSTEM TEMPERATURE: ";TAB(45)FN X(SNT + SKYTEMP);"
      K"
1540  LPRINT
1550  LPRINT "EFFECTIVE RADIATED POWER LEVEL: "; TAB(45)  FN X(PT);" DBM"
1560  LPRINT "MINUS PATH LOSS: "; TAB(45)  FN X( - ALPHA);" DB"
1570  LPRINT "PLUS RECEIVE ANTENNA GAIN: "; TAB(45); FN X(GX);" DBI"
1580  LPRINT "RECEIVED SIGNAL LEVEL: "; TAB(45)FN X(RL);" DBM"
1590  LPRINT : PRINT "RECEIVER NOISE LEVEL: "; TAB(45)  FN X(NOISE);" DBM"
1600  LPRINT "SIGNAL TO NOISE RATIO: "; TAB(45)  FN X(STN);" DB";
1610  IF STN < 0 THEN  LPRINT " (BELOW THE NOISE)";
1630 LPRINT:LPRINT:LPRINT
1640  RETURN
1700  PRINT: CLS: PRINT
1710  INPUT "WHAT IS THE COLD SKY TEMPERATURE IN DEGREES K?";COLDSKY
1720  PRINT : PRINT "WHAT IS THE MOON TEMPERATURE IN DEGREES K"
1730  INPUT "   (150 DEG K WAS INITIAL VALUE)? ";MNTEMP
1740  RETURN
1800 REM PRINTER INITIALIZATION, IF USED
1810  RETURN
```

EME SYSTEM CALCULATOR (Apple)

```
10  REM  EME SYSTEM CALCULATIONS
20  REM  BY KC6A AND N6NB
30  REM  (C) 1983 BY HAYDEN BOOK CO.,INC.
40 D$ =  CHR$ (4)
50 LP$ = D$ + "PR#1"
52 NWPAGE$ =  CHR$ (12)
60 SCN$ = D$ + "PR#0"
70  DEF  FN LG(X) =  LOG (X) /  LOG (10)
80  DEF  FN AL(X) = 10 ^ (X / 10)
90  DEF  FN NT(X) = 293 * (10 ^ (X / 10) - 1)
100 COLDSKY = 50: REM  COLD SKY TEMP IN K
110 MNTEMP = 150: REM  MOON TEMP IN DEG K
120  DEF  FN X(X) =  INT (10 * X) / 10
130  HOME : PRINT : PRINT : INVERSE : GOSUB 160: GOSUB 160
140  HTAB (5): PRINT "    EME SYSTEM CALCULATOR     "
150  GOSUB 160: GOSUB 160: NORMAL : GOTO 170
160  HTAB (5): PRINT "                              ": RETURN
170  PRINT : PRINT " THIS PROGRAM CALCULATES THE RECEIVED"
180  PRINT "SIGNAL-TO-NOISE RATIO FOR AN EME SYSTEM"
190  PRINT
200  FOR ITM = 1 TO 12
210  ON ITM GOSUB 800,820,850,880,920,970,1000,1050,1080,1110,1140,1300
220  NEXT ITM
230  HOME : PRINT : INPUT "DO YOU WISH TO CHANGE ANYTHING? (Y/N)";C$
240  IF C$ <  > "Y" THEN 410
250  HOME : PRINT : PRINT " 1.  FREQUENCY"
260  PRINT " 2.  TRANSMIT POWER"
270  PRINT " 3.  TRANSMIT LINE LOSS"
280  PRINT " 4.  RX ANT LINE LOSS"
290  PRINT " 5.  RX PREAMP #1"
300  PRINT " 6.  RX PREAMP #1 LINE LOSS"
310  PRINT " 7.  RX PREAMP #2"
320  PRINT " 8.  RX PREAMP #2 LINE LOSS"
330  PRINT " 9.  RX CONVERTER"
340  PRINT "10.  RX BANDWIDTH"
350  PRINT "11.  ANTENNA"
360  PRINT "12.  HARD COPY"
362  PRINT "13.  SKY TEMPERATURE"
370  PRINT : INPUT "WHICH ITEM DO YOU WISH TO CHANGE? ";ITM
380  IF ITM > 13 THEN 230
390  ON ITM GOSUB 800,830,850,890,920,970,1000,1050,1080,1110,1150,1300,1700
400  GOTO 230
410  REM  CALCULATE TRANSMIT SIGNAL LEVEL
420  IF RD$ = "Y" THEN GX = 20 *  FN LG(FRQ) + 20 *  FN LG(RD) - 52.6
430  IF TD$ = "Y" THEN GT = 20 *  FN LG(FRQ) + 20 *  FN LG(TD) - 52.6
440 PT = 10 *  FN LG(PO) + 30 - TL + GT
450 LBTW2 = 10 *  FN LG(27000) - GT: REM  LOG TX BEAMWIDTH^2 (DEG)
460 BTW =  FN AL(LBTW2 / 2): REM  TX BW
470 BRW =  FN AL((10 *  FN LG(27000) - GX) / 2): REM  RX BEAMWIDTH
480 RS =  FN LG(.25) - (10 *  FN LG(27000) - GX): IF RS > 0 THEN RS = 0: REM  AREA
490  REM  CALCULATE PATH ATTENUATION
500 ALPHA = 209.7 + 20 *  FN LG(FRQ): REM  PATH LOSS
510 RL = PT - ALPHA + GX: REM  RECEIVED SIGNAL LEVEL
520  REM  CALCULATE SYSTEM NOISE TEMP
530 M1 =  FN AL(L1):H1 =  FN AL(G1):O1 =  FN NT(N1)
540 M2 =  FN AL(L2):H2 =  FN AL(G2):O2 =  FN NT(N2)
550 M3 =  FN AL(L3):O3 =  FN NT(N3)
560 SKYTEMP = COLDSKY +  FN AL(RS) * (MNTEMP - COLDSKY)
570 SNT = 293 * (M1 - 1) + (O1 * M1) + 293 * M1 / H1 * (M2 - 1) + O2 * (M1 * M2 /
H1) + 293 * (M3 - 1) * (M1 * M2 / (H1 * H2)) + O3 * (M1 * M2 * M3 / (H1 * H2))
580 NOISE = 10 *  FN LG SNT + SKYTEMP) + 10 *  FN LG(BW * 800) - 198.6
590 STN = RL - NOISE: REM  SIGNAL TO NOISE LEVEL IN DB
600  HOME
602  PRINT "FREQUENCY="; TAB( 27)FRQ;" MHZ"
610  PRINT "RECEIVER NOISE TEMPERATURE ="; TAB( 27); FN X(SNT)
620  PRINT "SKY TEMPERATURE ="; TAB( 27) FN X(SKYTEMP);" K"
630  PRINT "SYSTEM NOISE TEMPERATURE ="; TAB( 27) FN X(SNT + SKYTEMP);" K"
```

```
640  PRINT "INPUT NOISE LEVEL ="; TAB( 27) FN X(NOISE);" DBM"
650  PRINT : PRINT "TRANSMIT ANTENNA GAIN = "; FN X(GT);" DBI"
660  PRINT  TAB( 8)"3DB BEAMWIDTH = "; FN X(BTW);" DEG"
670  PRINT "RECEIVE ANTENNA GAIN = "; FN X(GX);" DBI"
680  PRINT  TAB( 8)"3DB BEAMWIDTH = "; FN X(BRW);" DEG"
690  PRINT
700  PRINT : PRINT "TRANSMIT LEVEL ="; TAB( 27) FN X(PT);" DBM"
710  PRINT "PATH LOSS ="; TAB( 27) FN X(ALPHA);" DB"
720  PRINT "RECEIVED SIGNAL ="; TAB( 27) FN X(RL);" DBM"
730  PRINT : PRINT "SIGNAL TO NOISE RATIO = "; FN X(STN);" DB";
740  IF STN < 0 THEN  PRINT " (BELOW NOISE)";
750  PRINT
760  IF HC$ = "Y" THEN  GOSUB 1320
770  PRINT : INPUT "DO YOU WANT TO RE-RUN THE PROGRAM? (Y/N)? ";RR$
780  IF RR$ = "Y" THEN 230
790  END
800  PRINT : INPUT "1. WHAT IS THE OPERATING FREQUENCY (MHZ)? ";FRQ
810  RETURN
820  HOME : PRINT  TAB( 5)"TRANSMITTER SPECIFICATIONS": PRINT
830  PRINT : INPUT "2. WHAT IS THE TRANSMITTER OUTPUT POWER (WATTS)? ";PO
840  RETURN
850  PRINT : PRINT "3. WHAT IS THE TRANSMITTER TO ANTENNA"
860  INPUT "   LINE LOSS IN DB? ";TL
870  RETURN
880  HOME : PRINT  TAB( 5)"RECEIVER SPECIFICATIONS": PRINT
890  PRINT : PRINT "4. WHAT IS THE LINE LOSS FROM THE"
900  INPUT "    ANTENNA TO THE 1ST PREAMP IN DB? ";L1
910  RETURN
920  PRINT : PRINT "5. WHAT IS THE GAIN OF THE 1ST"
930  INPUT "   PREAMP IN DB? ";G1
940  PRINT : PRINT "   WHAT IS THE NOISE FIGURE OF THE"
950  INPUT "   1ST PREAMP IN DB? ";N1
960  RETURN
970  PRINT : PRINT "6. WHAT IS THE LINE LOSS FROM THE 1ST"
980  INPUT "   PREAMP TO THE 2ND PREAMP IN DB? ";L2
990  RETURN
1000  PRINT : PRINT "7. WHAT IS THE GAIN OF THE 2ND"
1010  INPUT "   PREAMP IN DB? ";G2
1020  PRINT : PRINT "   WHAT IS THE NOISE FIGURE OF THE"
1030  INPUT "   2ND PREAMP IN DB? ";N2
1040  RETURN
1050  PRINT : PRINT "8. WHAT IS THE LINE LOSS FROM THE 2ND"
1060  INPUT "   PREAMP TO THE CONVERTER IN DB? ";L3
1070  RETURN
1080  PRINT : PRINT "9. WHAT IS THE NOISE FIGURE OF THE"
1090  INPUT "   RECEIVE CONVERTER IN DB? ";N3
1100  RETURN
1110  PRINT : PRINT "10. WHAT IS THE RECEIVE BANDWIDTH"
1120  INPUT "    IN KHZ? ";BW
1130  RETURN
1140  HOME : PRINT  TAB( 5)"ANTENNA PARAMETERS": PRINT
1150  INPUT "11. IS THE RECEIVE ANTENNA A DISH (Y/N)? ";RD$
1160  PRINT
1170  IF RD$ = "Y" THEN 1200
1180  PRINT "    WHAT IS THE RECEIVE ANTENNA GAIN"
1190  INPUT "    OVER AN ISOTROPIC SOURCE IN DB? ";GX: GOTO 1210
1200  INPUT "WHAT IS THE DIAMETER IN FEET? ";RD
1210  PRINT : INPUT "IS THE TRANSMIT ANTENNA THE SAME (Y/N)? ";TT$
1220  IF TT$ = "Y" THEN GT = GX:TD$ = RD$:TD = RD: GOTO 1290
1230  PRINT : INPUT "    IS THE TRANSMIT ANTENNA A DISH? ";TD$
1240  PRINT
1250  IF TD$ = "Y" THEN 1280
1260  PRINT "    WHAT IS THE TRANSMIT ANTENNA GAIN"
1270  INPUT "    OVER AN ISOTROPIC SOURCE IN DB? ";GT: GOTO 1290
1280  INPUT "    WHAT IS THE DIAMETER IN FEET? ";TD
1290  RETURN
1300  PRINT : INPUT "DO YOU WANT HARD COPY (Y/N)? ";HC$
1302  IF HC$ = "Y" THEN  GOSUB 1800: REM  INIT PRINTER, IF NEEDED
1310  RETURN
1320  PRINT LP$
```

```
1330  PRINT  TAB( 20)"EME SYSTEM CALCULATOR"
1340  PRINT : PRINT  TAB( 17)"TRANSMITTER SPECIFICATIONS"
1341  PRINT
1342  PRINT "FREQUENCY ="; SPC( 34)FRQ;" MHZ"
1350  PRINT : PRINT "TRANSMITTER OUTPUT POWER:"; SPC( 20)PO;" WATTS"
1360  PRINT "TRANSMITTER TO ANTENNA LINE LOSS:"; SPC( 12)TL;" DB"
1370  PRINT "TRANSMIT ANTENNA: ";
1380  IF TD$ = "Y" THEN  PRINT TD;" FOOT DISH. ";
1390  PRINT "GAIN = "; FN X(GT);" DBI."
1392  PRINT  TAB( 18)"3DB BEAMWIDTH = "; FN X(BTW);" DEG."
1394  PRINT  TAB( 18)"APPROX POINTING ACCURACY = "; FN X(BTW / 6);" DEG."
1400  PRINT : PRINT  TAB( 19)"RECEIVER SPECIFICATIONS"
1410  PRINT : PRINT  TAB( 12)"LINE LOSS    GAIN    NF"
1420  PRINT "PREAMP 1     ";L1;" DB";: HTAB (22): PRINT G1;" DB";: HTAB (30): PRINT
 N1;" DB"
1430  PRINT "PREAMP 2     ";L2;" DB";: HTAB (22): PRINT G2;" DB";: HTAB (30): PRINT
 N2;" DB"
1440  PRINT "CONVERTER    ";L3;" DB";: HTAB (30): PRINT N3;" DB"
1442  PRINT "RECEIVER BANDWIDTH:"; SPC( 16)BW;" KHZ"
1450  PRINT "RECEIVE ANTENNA: ";
1460  IF RD$ = "Y" THEN  PRINT RD;" FOOT DISH. ";
1470  PRINT "GAIN = "; FN X(GX);" DBI."
1472  PRINT  TAB( 18)"3DB BEAMWIDTH = "; FN X(BRW);" DEG."
1474  PRINT  TAB( 18)"APPROX POINTING ACCURACY = "; FN X(BRW / 6);" DEG."
1480  PRINT "RECEIVE SYSTEM TEMPERATURE:"; SPC( 18) FN X(SNT);" K"
1490  PRINT
1500  PRINT "BASED ON A COLD SKY TEMPERATURE OF ";COLDSKY;" DEGREES K"
1510  PRINT "AND A MOON TEMPERATURE OF ";MNTEMP;" DEGREES K,"
1520  PRINT "THE APPROXIMATE ANTENNA TEMPERATURE ="; SPC( 8); FN X(SKYTEMP);" K"
1530  PRINT "AND THE OVERALL SYSTEM TEMPERATURE ="; SPC( 9); FN X(SNT + SKYTEMP);"
 K"
1540  PRINT
1550  PRINT "EFFECTIVE RADIATED POWER LEVEL:"; SPC( 14) FN X(PT);" DBM"
1560  PRINT "MINUS PATH LOSS:"; SPC( 28) FN X( - ALPHA);" DB"
1570  PRINT "PLUS RECEIVE ANTENNA GAIN:"; SPC( 19); FN X(GX);" DBI"
1580  PRINT "RECEIVED SIGNAL LEVEL ="; SPC( 22); FN X(RL);" DBM"
1590  PRINT : PRINT "RECEIVER NOISE LEVEL ="; SPC( 23) FN X(NOISE);" DBM"
1600  PRINT "SIGNAL TO NOISE RATIO ="; SPC( 22) FN X(STN);" DB";
1610  IF STN < 0 THEN  PRINT "  (BELOW THE NOISE)";
1620  PRINT NWPAGE$
1630  PRINT SCN$
1640  RETURN
1700  HOME : PRINT
1710  INPUT "WHAT IS THE COLD SKY TEMPERATURE IN DEGREES K?";COLDSKY
1720  PRINT : PRINT "WHAT IS THE MOON TEMPERATURE IN DEGREES K"
1730  INPUT "   (150 DEG K WAS INITIAL VALUE)? ";MNTEMP
1740  RETURN
1800  PRINT LP$
1802  REM  PUT PRINTER INIT HERE IF NEEDED
1809  PRINT SCN$
1810  RETURN
```

EME SYSTEM CALCULATOR (Commodore 64)

```
10 REM EME SYSTEM CALCULATIONS (C-64)
20 REM BY KC6A AND N6NB.  (C) 1983 BY HAYDEN BOOK CO.,INC.
30 OPEN130,2,2,CHR$(136)+CHR$(1):REM RS232 1200-BAUD PRINTER INIT
60 DEF FNX(X)=INT (10*X)/10
70 DEF FNLG(X)=LOG (X)/ LOG (10)
80 DEF FNAL(X)=10^(X/10)
90 DEF FNNT(X)=293*(10^(X/10)-1)
100 COLDSKY=50:REM COLD SKY TEMP IN K
110 MNTEMP=150:REM MOON TEMP IN DEG K
120 CLS$=CHR$(147):REM CLEAR-SCREEN COMMAND
```

```
130 PRINT CLS$:PRINT:PRINT
140 PRINT TAB(9)"EME SYSTEM CALCULATOR"
160 PRINT
170 PRINT:PRINT "THIS PROGRAM CALCULATES THE RECEIVED"
180 PRINT "SIGNAL TO NOISE RATIO FOR AN EME SYSTEM"
190 PRINT
200 FOR ITM=1 TO 12
210 ON ITM GOSUB 800,820,850,880,920,970,1000,1050,1080,1110,1140,1300
220 NEXT ITM
230 PRINT CLS$:PRINT:INPUT "DO YOU WISH TO CHANGE ANYTHING (Y/N)";C$
240 IF C$<>"Y" THEN 410
250 PRINT CLS$:PRINT:PRINT " 1.  FREQUENCY"
260 PRINT " 2.  TRANSMIT POWER"
270 PRINT " 3.  TRANSMIT LINE LOSS"
280 PRINT " 4.  RX ANT LINE LOSS"
290 PRINT " 5.  RX PREAMP #1"
300 PRINT " 6.  RX PREAMP #1 LINE LOSS"
310 PRINT " 7.  RX PREAMP #2"
320 PRINT " 8.  RX PREAMP #2 LINE LOSS"
330 PRINT " 9.  RX CONVERTER"
340 PRINT "10.  RX BANDWIDTH"
350 PRINT "11.  ANTENNA"
360 PRINT "12.  HARD COPY"
362 PRINT "13.  SKY TEMPERATURE"
370 PRINT:INPUT "WHICH ITEM DO YOU WISH TO CHANGE ";ITM
380 IF ITM>13 THEN 230
390 ON ITM GOSUB 800,830,850,890,920,970,1000,1050,1080,1110,1150,1300,1700
400 GOTO 230
410 REM CALCULATE TRANSMIT SIGNAL LEVEL
420 IF RD$="Y" THEN GX=20*FNLG(FRQ)+20*FNLG(RD)-52.6
430 IF TD$="Y" THEN GT=20*FNLG(FRQ)+20*FNLG(TD)-52.6
440 PT=10*FNLG(PO)+30-TL+GT
450 LBTW2=10*FNLG(27000)-GT:REM LOG TX BEAMWIDTH^2 (DEG)
460 BTW=FNAL(LBTW2/2):REM TX BW
470 BRW=FNAL((10*FNLG(27000)-GX)/2):REM RX BEAMWIDTH
480 RS=FNLG(.25)-(10*FNLG(27000)-GX):IF RS>0 THEN RS=0:REM AREA
490 REM CALCULATE PATH ATTENUATION
500 ALPHA=209.7+20*FNLG(FRQ):REM PATH LOSS
510 RL=PT-ALPHA+GX:REM RECEIVED SIGNAL LEVEL
520 REM CALCULATE SYSTEM NOISE TEMP
530 M1=FNAL(L1):H1=FNAL(G1):O1=FNNT(N1)
540 M2=FNAL(L2):H2=FNAL(G2):O2=FNNT(N2)
550 M3=FNAL(L3):O3=FNNT(N3)
560 SKYTEMP=COLDSKY+ FNAL(RS)*(MNTEMP-COLDSKY)
570 SNT=293*(M1-1)+(O1*M1)+293*M1/H1*(M2-1)+O2*(M1*M2/H1)
572 SNT=SNT+293*(M3-1)*(M1*M2/(H1*H2))+O3*(M1*M2*M3/(H1*H2))
580 NOISE=10*FNLG(SNT+SKYTEMP)+10*FNLG(BW*800)-198.6
590 S9=RL-NOISE:REM SIGNAL TO NOISE LEVEL IN DB
600 PRINT CLS$
602 PRINT "FREQUENCY=";TAB(27)FRQ;" MHZ"
610 PRINT "RCVR NOISE TEMPERATURE =";TAB(27)FNX(SNT);" K"
620 PRINT "SKY TEMPERATURE =";TAB(27) FNX(SKYTEMP);" K"
630 PRINT "SYSTEM NOISE TEMPERATURE =";TAB(27) FNX(SNT+SKYTEMP);" K"
640 PRINT "INPUT NOISE LEVEL =";TAB(27) FNX(NOISE);" DBM"
650 PRINT:PRINT "TRANSMIT ANTENNA GAIN="; FNX(GT);" DBI"
660 PRINT TAB(8)"3DB BEAMWIDTH="; FNX(BTW);" DEG"
670 PRINT "RECEIVE ANTENNA GAIN="; FNX(GX);" DBI"
680 PRINT TAB(8)"3DB BEAMWIDTH="; FNX(BRW);" DEG"
690 PRINT
700 PRINT:PRINT "TRANSMIT LEVEL =";TAB(27) FNX(PT);" DBM"
710 PRINT "PATH LOSS =";TAB(27) FNX(ALPHA);" DB"
720 PRINT "RECEIVED SIGNAL =";TAB(27) FNX(RL);" DBM"
730 PRINT:PRINT "SIGNAL TO NOISE RATIO="; FNX(S9);" DB";
740 IF S9<0 THEN PRINT " (BELOW NOISE)";
750 PRINT
760 IF LP$="Y" THEN GOSUB 1330
770 PRINT:PRINT "DO YOU WANT TO RE-RUN THE PROGRAM (Y/N) ":INPUT RR$
780 IF RR$="Y" THEN 230
790 FOR J=1 TO 1999:PRINT;:NEXT J:CLOSE130:END
800 PRINT:PRINT "1. WHAT IS THE OPERATING FREQUENCY (MHZ) ":INPUT FRQ
```

```
810 RETURN
820 PRINT:PRINT:PRINT TAB(6)"TRANSMITTER SPECIFICATIONS":PRINT
830 PRINT:PRINT "2. WHAT IS THE TRANSMITTER OUTPUT POWER (WATTS) ":INPUT PO
840 RETURN
850 PRINT:PRINT "3. WHAT IS THE TRANSMITTER TO ANTENNA"
860 INPUT "   LINE LOSS IN DB ";TL
870 RETURN
880 PRINT:PRINT:PRINT TAB(6)"RECEIVER SPECIFICATIONS":PRINT
890 PRINT:PRINT "4. WHAT IS THE LINE LOSS FROM THE"
900 PRINT "    ANTENNA TO THE 1ST PREAMP IN DB ":INPUT L1
910 RETURN
920 PRINT:PRINT "5. WHAT IS THE GAIN OF THE 1ST"
930 INPUT "   PREAMP IN DB ";G1
940 PRINT:PRINT "   WHAT IS THE NOISE FIGURE OF THE"
950 INPUT "   1ST PREAMP IN DB ";N1
960 RETURN
970 PRINT:PRINT "6. WHAT IS THE LINE LOSS FROM THE 1ST"
980 PRINT "   PREAMP TO THE 2ND PREAMP IN DB ":INPUT L2
990 RETURN
1000 PRINT:PRINT "7. WHAT IS THE GAIN OF THE 2ND"
1010 INPUT "   PREAMP IN DB ";G2
1020 PRINT:PRINT "   WHAT IS THE NOISE FIGURE OF THE"
1030 INPUT "   2ND PREAMP IN DB ";N2
1040 RETURN
1050 PRINT:PRINT "8. WHAT IS THE LINE LOSS FROM THE 2ND"
1060 INPUT "   PREAMP TO THE CONVERTER IN DB ";L3
1070 RETURN
1080 PRINT:PRINT "9. WHAT IS THE NOISE FIGURE OF THE"
1090 INPUT "   RECEIVE CONVERTER IN DB ";N3
1100 RETURN
1110 PRINT:PRINT "10. WHAT IS THE RECEIVE BANDWIDTH"
1120 INPUT "    IN KHZ ";BW
1130 RETURN
1140 PRINT CLS$
1145 PRINT:PRINT TAB(6)"ANTENNA PARAMETERS":PRINT
1150 PRINT "11. IS THE RECEIVE ANTENNA A DISH (Y/N) ":INPUT RD$
1160 PRINT
1170 IF RD$="Y" THEN 1200
1180 PRINT "    WHAT IS THE RECEIVE ANTENNA GAIN"
1190 PRINT "    OVER AN ISOTROPIC SOURCE IN DB ":INPUT GX:GOTO 1210
1200 PRINT "WHAT IS THE DIAMETER IN FEET ":INPUT RD
1210 PRINT:PRINT "IS THE TRANSMIT ANTENNA THE SAME (Y/N) ":INPUT TT$
1220 IF TT$="Y" THEN GT=GX:TD$=RD$:TD=RD:GOTO 1290
1230 PRINT:PRINT "    IS THE TRANSMIT ANTENNA A DISH ":INPUT TD$
1240 PRINT
1250 IF TD$="Y" THEN 1280
1260 PRINT "    WHAT IS THE TRANSMIT ANTENNA GAIN"
1270 PRINT "    OVER AN ISOTROPIC SOURCE IN DB ":INPUT GT:GOTO 1290
1280 PRINT "    WHAT IS THE DIAMETER IN FEET ":INPUT TD
1290 RETURN
1300 PRINT:PRINT "DO YOU WANT HARD COPY (Y/N) ":INPUT LP$
1310 RETURN
1330 PRINT#130,"";SPC(20)"EME SYSTEM CALCULATOR"
1340 PRINT#130,"":PRINT#130,"";SPC(17)"TRANSMITTER SPECIFICATIONS"
1341 PRINT#130,""
1342 PRINT#130,"FREQUENCY:";SPC(35)FRQ;" MHZ"
1350 PRINT#130,"":PRINT#130,"TRANSMITTER OUTPUT POWER:";SPC(20)PO;" WATTS"
1360 PRINT#130,"TRANSMITTER TO ANTENNA LINE LOSS:";SPC(12)TL;" DB"
1370 PRINT#130,"TRANSMIT ANTENNA:"
1380 IF TD$="Y" THEN PRINT#130,"     ";TD;" FOOT DISH--";
1390 PRINT#130,"     GAIN:";SPC(14)FNX(GT);" DBI."
1392 PRINT#130,"";SPC(18)"3DB BEAMWIDTH:";SPC(13)FNX(BTW);" DEG."
1394 PRINT#130,"";SPC(18)"APPROX POINTING ACCURACY:";"  "FNX(BTW/6);" DEG."
1400 PRINT#130,"":PRINT#130,"";SPC(19)"RECEIVER SPECIFICATIONS"
1410 PRINT#130,"":PRINT#130,"";SPC(12)"LINE LOSS    GAIN    NF"
1420 PRINT#130, "PREAMP 1    ";L1;" DB";"    ";G1;" DB";SPC(3)N1;" DB"
1430 PRINT#130, "PREAMP 2    ";L2;" DB";"    ";G2;" DB";SPC(3)N2;" DB"
1440 PRINT#130, "CONVERTER   ";L3;" DB";"              ";N3;" DB"
1442 PRINT#130, "RECEIVER BANDWIDTH:";SPC(26)BW;" KHZ"
```

```
1450 PRINT#130, "RECEIVE ANTENNA:"
1460 IF RD$="Y" THEN PRINT#130,"     "; RD;" FOOT DISH--";
1470 PRINT#130, "     GAIN:";SPC(14)FNX(GX);" DBI."
1472 PRINT#130,"";SPC(18)"3DB BEAMWIDTH:";SPC(13)FNX(BRW);" DEG."
1474 PRINT#130,"";SPC(18)"APPROX POINTING ACCURACY:";"  ";FNX(BRW/6);" DEG."
1480 PRINT#130, "RECEIVE SYSTEM TEMPERATURE:";SPC(18)FNX(SNT);" K"
1490 PRINT#130,""
1500 PRINT#130, "BASED ON A COLD SKY TEMPERATURE OF ";
1502 PRINT#130,"";SPC(10)COLDSKY;" DEGREES K"
1510 PRINT#130, "AND A MOON TEMPERATURE OF ";SPC(19)MNTEMP;" DEGREES K,"
1520 PRINT#130, "THE APPROXIMATE ANTENNA TEMPERATURE:";SPC(9)FNX(SKYTEMP);" K"
1530 PRINT#130, "AND THE OVERALL SYSTEM TEMPERATURE:";
1532 PRINT#130,"";SPC(10)FNX(SNT+SKYTEMP);" K"
1540 PRINT#130,""
1550 PRINT#130, "EFFECTIVE RADIATED POWER LEVEL:";SPC(14) FNX(PT);" DBM"
1560 PRINT#130, "MINUS PATH LOSS:";SPC(29)FNX(-ALPHA);" DB"
15/0 PRINT#130, "PLUS RECEIVE ANTENNA GAIN:";SPC(19)FNX(GX);" DBI"
1580 PRINT#130, "RECEIVED SIGNAL LEVEL:";SPC(23)FNX(RL);" DBM"
1590 PRINT#130,"";PRINT "RECEIVER NOISE LEVEL:";SPC(16)FNX(NOISE);" DBM"
1600 PRINT#130, "SIGNAL TO NOISE RATIO:";SPC(23)FNX(S9);" DB";
1610 IF S9<0 THEN PRINT#130, " (BELOW THE NOISE)";
1630 PRINT#130,"":PRINT#130,"":PRINT#130,""
1640 RETURN
1700 PRINT:PRINT CLS$:PRINT
1710 PRINT "WHAT IS THE COLD SKY TEMPERATURE IN DEGREES K?":INPUT COLDSKY
1720 PRINT:PRINT "WHAT IS THE MOON TEMPERATURE IN DEGREES K"
1730 PRINT "  (150 DEG K WAS INITIAL VALUE)? ":INPUT MNTEMP
1740 RETURN
```

THE MOONTRACKER

What it does

This classic program, originally written by Lance Collister (and modified by many others over the years), tracks the moon across the sky, printing out elevation-azimuth tracking data—customized to your location—for as many days as you wish. This version includes several new features, most notably the calendar logic to print results for an unlimited number of days. Given your geographic coordinates, the date, and some other information, this program goes to work.

The MOONTRACKER also computes the moon's declination and the Greenwich Hour Angle (which indicates the longitude where the moon is directly overhead at that particular time). The program displays the "universal" (i.e., European), W/VE and VK/ZL/JA "windows" to the right of the declination.

Setup instructions

Despite the complexity of the mathematical formulas, this program will work on an amazing variety of computers. In addition to the normal listings for Apple, Commodore, and systems running versions of Microsoft BASIC (TRS-80 and CP/M), we have included a listing of a special version of this program for the Timex/Sinclair 1000. Just about any computer with at least 8K of free memory will run this program.

There is one way in which you may wish to customize the program. By revising line 210, you can set it up to provide for automatic entry of your latitude and longitude whenever you enter your own call sign. You could also add more lines immediately below 210 to test for the call signs

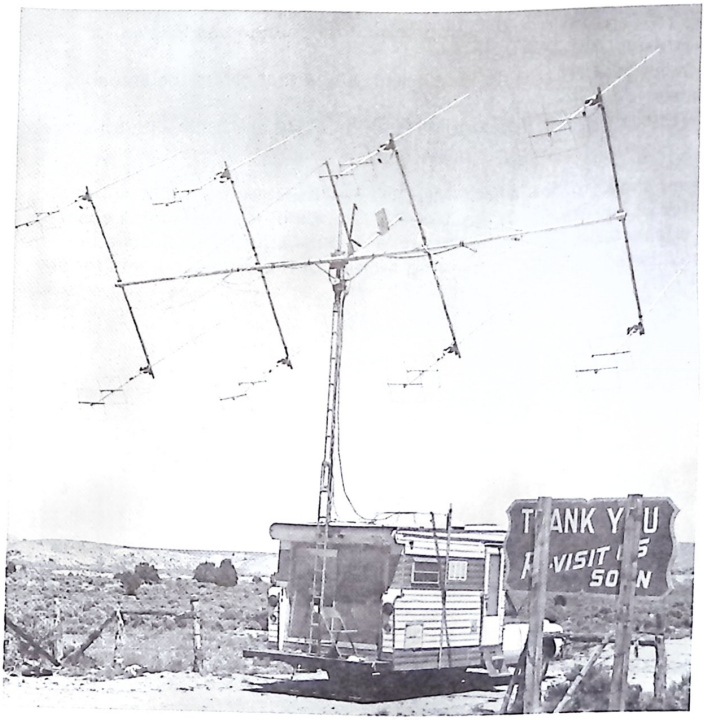
and automatically enter the coordinates of other EME enthusiasts for whom you provide moontracking data. The variable L5 is the latitude in degrees, and U5 is the minutes; L6 is the longitude in degrees, while U6 is the minutes.

Operating notes

The MOONTRACKER first asks for your call sign; if you enter a call for which it is customized, the questions about latitude and longitude are bypassed.

Now you have some choices to make. You can have a hard-copy printout (a hard copy is very useful with the detailed information this program computes). Also, you can specify a printing increment from 1 to 60 minutes. Just to see the program in action, type 60. For serious EME work, you'll probably want computations at 15-minute intervals or perhaps even more frequently.

You can also have the program print tracking data only when the moon is below a specific elevation angle. This is useful for those who cannot point their antennas high in the sky for one reason or another.



This is N6NB's portable moonbounce array of eight quagis. The truck is positioned exactly on the Utah-Nevada state line; it could be driven from one state to the other with this moonbounce array fully assembled to hand out contacts with these two rare states via two-meter EME. But whether you're into mobile moonbounce or just doing it at home, the MOONTRACKER program will provide full tracking data.

Finally, you enter the starting date (use commas to separate the month, day and year) and specify the number of days for which you want data. You must enter all four digits of the year; if you just typed "84" for 1984, the program would try to compute moontracking data for the year 84 A.D., although the results would not be accurate due to changes in the calendar since then.

Provided your printer is well stocked with paper (and you're patient—this program takes time to run), you can tell it to print out data for a whole year if you like.

The routines to count the days and determine when an old year ends and a new one begins (you can start with a date in late December; the program knows when New Year's Day is) are new in this version of the MOONTRACKER. The program also figures out if it's a leap year, and adjusts itself accordingly.

After you've typed all of the formulas and run the program, of course, you will probably be tempted to rush outside and see where the moon really is. If you've typed it correctly, it will be where the program says it will be. This program's output closely matches the data published in the *Nautical Almanac* when the coordinates used by the *Almanac* are entered.

Modifications for CP/M

The TRS-80 and CP/M listing should be changed as follows for CP/M systems running BASIC-80:

1. Delete line 70 and replace it with a line that clears the screen on your system;

2. Replace the left brackets ("[") in lines 940 and 1250 with up-arrows ("^").

Modifications for Timex

A simplified version of the MOONTRACKER program that will run on a Timex computer (with the 16K RAM module) is listed in this chapter. It does not include the multiple-day printout feature, but it does perform the calculations properly, displaying a chart of moontracking data for any day.

Sample printout of MOONTRACKER (TRS-80).

MOON TRACKING PROGRAM

DATA COMPUTED FOR KC6A

LATITUDE: 33 - 47.1 LONGITUDE: 118 - 5.95

U = EUROPEAN WINDOW. W = W-VE WINDOW. J = JA-VK-ZL WINDOW

POSITION OF MOON ON 1 / 1 / 1984 0 - 2400 UTC FOLLOWS:

UTC	AZ	EL	GHA	DEC
1400	121.5	3.3	49.7	-23.7
1430	125.9	8.3	57.0	-23.7
1500	130.6	12.9	64.2	-23.8
1530	135.8	17.3	71.4	-23.8
1600	141.4	21.2	78.7	-23.8
1630	147.6	24.7	85.9	-23.9
1700	154.2	27.6	93.1	-23.9
1730	161.3	29.8	100.4	-24.0
1800	168.8	31.4	107.6	-24.0
1830	176.4	32.1	114.8	-24.0
1900	184.3	32.0	122.1	-24.1
1930	191.9	31.1	129.3	-24.1
2000	199.4	29.5	136.5	-24.1
2030	206.4	27.1	143.8	-24.2
2100	212.9	24.1	151.0	-24.2
2130	218.9	20.6	158.2	-24.2
2200	224.4	16.5	165.5	-24.3
2230	229.5	12.1	172.7	-24.3
2300	234.1	7.4	179.9	-24.3
2330	238.4	2.4	187.2	-24.4

POSITION OF MOON ON 1 / 2 / 1984 0 - 2400 UTC FOLLOWS:

UTC	AZ	EL	GHA	DEC
1500	123.5	3.6	51.4	-25.1
1530	127.9	8.4	58.6	-25.1
1600	132.7	13.0	65.9	-25.1
1630	137.9	17.3	73.1	-25.1
1700	143.5	21.0	80.3	-25.1
1730	149.7	24.3	87.6	-25.1
1800	156.3	27.1	94.8	-25.1
1830	163.3	29.1	102.1	-25.1
1900	170.7	30.5	109.3	-25.1
1930	178.3	31.0	116.5	-25.2
2000	185.9	30.8	123.7	-25.2
2030	193.4	29.8	131.0	-25.2
2100	200.6	28.0	138.2	-25.2
2130	207.4	25.6	145.4	-25.2
2200	213.8	22.5	152.7	-25.2
2230	219.6	18.9	159.9	-25.2
2300	225.0	14.9	167.1	-25.2

MOONTRACKER (TRS-80 and CP/M)

```
10 REM MOON TRACKING PROGRAM (TRS-80 VERSION)
20 REM ADAPTED FROM WA1JXN'S PROGRAM BY KC6A AND N6NB.  FOR EASE OF
30 REM DATA INPUT, PRINTOUT OF AN UNLIMITED NUMBER OF CONSECUTIVE
40 REM DAYS REQUIRES ONLY ONE ENTRY DATE.  LATITUDE/LONGITUDE INPUT
50 REM IS AUTOMATIC FOR PRIMARY USERS, AND OUTPUT GOES TO BOTH CRT
60 REM AND LINE PRINTER (WITH FULL PAGINATION).
70 CLEAR 250:CLS
80 DIM W(12):REM CALENDAR INFORMATION (LEAP YEAR COMPUTED LATER)
90 W(1)=31:W(2)=28:W(3)=31:W(4)=30:W(5)=31:W(6)=30
100 W(7)=31:W(8)=31:W(9)=30:W(10)=31:W(11)=30:W(12)=31
110 DEF FNA(X)=INT(X*D5*10+.5)/10
120 DEF FNB(X)=(X-INT(X))*P5
130 P5=2*3.14159
140 D5=360/P5
150 R5=P5/360:FF=10
160 F1$="    UTC          AZ          EL          GHA          DEC"
170 F2$="    ####         ###.#       ##.#        ###.#        +###.# !"
180 CLS:PRINT:PRINT:PRINT TAB(20)"MOON TRACKING PROGRAM"
190 PRINT:PRINT:INPUT "WHAT IS YOUR NAME OR CALL SIGN ";Q$
200 PRINT:INPUT "DO YOU WANT HARD COPY (Y/N) ";LP$
210 IF Q$="KC6A" THEN L5=33:U5=47.1:L6=118:U6=5.95:GOTO 250
220 REM PUT YOUR CALL AND COORDINATES HERE
230 INPUT "WHAT IS THE LATITUDE IN DEGREES, MINUTES";L5,U5
240 INPUT "WHAT IS THE LONGITUDE IN DEGREES, MINUTES";L6,U6
250 X5=L5:X6=L6
260 L5=(L5+U5/60)*R5:L6=(L6+U6/60)*R5
270 PRINT
280 INPUT "WHAT IS THE DESIRED PRINTING INCREMENT IN MINUTES (1-60)";I
290 PRINT:PRINT"MOON TRACKING DATA COMPUTED FOR ";Q$
300 PRINT:PRINT "LATITUDE: ";X5;"-";U5;" LONGITUDE: ";X6;"-";U6
310 PRINT:PRINT "IF YOU ONLY WANT A PRINTOUT FOR TIMES WHEN THE MOON IS NEAR"
320 PRINT "THE HORIZON ENTER 'H' (OTHERWISE HIT <RETURN>) ";B$
330 I6=100
340 IF B$="H" THEN PRINT:INPUT "BELOW WHAT ELEV. IN DEGREES DO YOU WANT PRINTOUT
    TO OCCUR ";I6
350 PRINT:INPUT "WHAT IS THE STARTING UTC DATE (MONTH,DAY,YEAR. E.G. 7,4,1984)";
    M,D,Y
360 PRINT:INPUT "FOR HOW MANY DAYS DO YOU WANT A PRINTOUT ";N
370 IF LP$<>"Y" THEN 440
380 LPRINT:LPRINT:LPRINT
390 LPRINT:LPRINT TAB(27)"MOON TRACKING PROGRAM"
400 LPRINT:LPRINT TAB(26)"DATA COMPUTED FOR ";Q$
410 LPRINT:LPRINT TAB(15)"LATITUDE: ";X5;"-";U5;" LONGITUDE: ";X6;"-"U6
420 LPRINT:LPRINT TAB(9)"U = EUROPEAN WINDOW. W = W-VE WINDOW. J = JA-VK-ZL WIND
    OW"
430 LPRINT:LPRINT:LN=13
440 E1=2400:B=0:GOTO 460
450 Y1=Y-(INT(Y/100)*100)
460 PRINT:PRINT "POSITION OF MOON ON ";M;"/";D;"/";Y;" ";B;"-";E1;" UTC FOLLOWS:
    "
470 PRINT:PRINT F1$
480 PRINT:PRINT
490 IF LP$<>"Y" THEN 530
500 IF LN>=56 THEN FF=70-LN:GOSUB 1850
510 LPRINT:LPRINT TAB(7)"POSITION OF MOON ON ";M;"/";D;"/";Y;" ";B;"-";E1;" UTC
    FOLLOWS:"
520 LPRINT:LPRINT F1$:LN=LN+4
530 I1=2
540 REM CALCULATION OF JULIAN DATE FOLLOWS
550 IF M>=3 THEN 610
560 IF INT((Y-1853)/4)<11 THEN C1=0:GOTO 580
570 C1=-1
580 J1=365*(Y-1853)+D+30*(M+9)+INT((M+10)/2)
590 J2=INT((Y-1853)/4)+1+C1
600 GOTO 690
610 IF INT ((Y-1852)/4)<11 THEN C1=0:GOTO 630
620 C1=-1
```

```
630 IF M=9 THEN 660
640 IF M=11 THEN 660
650 C2=0:GOTO 670
660 C2=1
670 J1=365*(Y-1852)+D+30*(M-3)+INT((M-2)/2)
680 J2=INT((Y-1852)/4)+C1+C2
690 J=J1+J2
700 REM (JULIAN DATE-2397547.5) FOR 0 HOURS UTC
710 T1=J-17472.5
720 D9=(B-INT(B/100)*100)+INT(B/100)*60
730 D6=(E1-INT(E1/100)*100)+INT(E1/100)*60
740 D7=D9-D6
750 D8=D7-I
760 IF D7>0 THEN 780
770 GOTO 800
780 IF D8>=0 THEN 1670
790 B=E1
800 T=(B-INT(B/100)*100)/1440+INT(B/100)/24
810 T5=T1+T
820 K1=FNB(.751213+.036601102*T5)
830 K2=FNB(.822513+.0362916457*T5)
840 K3=FNB(.995766+.00273777852*T5)
850 K4=FNB(.974271+.0338631922*T5)
860 K5=FNB(.0312525+.0367481957*T5)
870 L8=K1+.658*R5*SIN(2*K4)+6.289*R5*SIN(K2)
880 L8=L8-1.274*R5*SIN(K2-2*K4)-.186*R5*SIN(K3)
890 L8=L8+.214*R5*SIN(2*K2)-.114*R5*SIN(2*K5)
900 L8=L8-.059*R5*SIN(2*K2-2*K4)-.057*R5*SIN(K2+K3-2*K4)
910 K6=K5+.6593*R5*SIN(2*K4)+6.2303*R5*SIN(K2)-1.272*R5*SIN(K2-2*K4)
920 L7=5.144*R5*SIN(K6)-.146*R5*SIN(K5-2*K4)
930 D1=COS(L7)*SIN(L8)*.397821+SIN(L7)*.917463
940 D1=ATN(D1/(SQR(1-D1[2])))
950 G1=50.5+((D1)/(.792))*D5
960 G2=80+((D1)/(.808))*D5
970 G3=141.5-((D1)*(.738)*D5)
980 G4=170.5-((D1)*(.857)*D5)
990 A2=COS(L7)*COS(L8)/COS(D1)
1000 A1=(COS(L7)*SIN(L8)*.917463-SIN(L7)*.397821)/COS(D1)
1010 A=ATN(A1/A2)
1020 REM IF YOU TYPED ALL THIS WITHOUT MAKING ANY MISTAKES,
1030 REM YOU BELONG IN A SECRETARIAL POOL!
1040 GOSUB 1300
1050 R1=A
1060 L1=.065709822*T1
1070 L=T*24*1.00274+6.64606+(L1-INT(L1/24)*24)
1080 L=(L-INT(L/24)*24)
1090 G=(L/24)*P5-R1
1100 IF G<P5 THEN 1130
1110 G=G-P5
1120 GOTO 1160
1130 IF G<0 THEN 1150
1140 GOTO 1160
1150 G=G+P5
1160 H=L6-G
1170 E3=COS(L5)*COS(H)*COS(D1)+SIN(D1)*SIN(L5)
1180 E2=SQR(1-(E3*E3))
1190 E=ATN(E3/E2)
1200 IF E<0 THEN 1620
1210 IF E>I6*R5 THEN 1620
1220 A2=SIN(D1)/(COS(L5)*COS(E))
1230 A2=A2-(SIN(L5)/COS(L5))*(SIN(E)/COS(E))
1240 A1=SIN(L5)*SIN(D1)+COS(L5)*COS(D1)*COS(H)
1250 A1=(SIN(H)*COS(D1))/(SQR(1-A1[2]))
1260 A=ATN(A1/A2)
1270 GOSUB 1300
1280 GOTO 1450
1290 REM
1300 IF A=0 THEN 1320
1310 GOTO 1360
1320 IF A2<0 THEN 1340
```

```
1330 GOTO 1440
1340 A=P5/2
1350 GOTO 1440
1360 IF A>0 THEN 1420
1370 IF A2<0 THEN 1400
1380 A=P5+A
1390 GOTO 1440
1400 A=P5+(A-P5/2)
1410 GOTO 1440
1420 IF A2>0 THEN 1440
1430 A=A+P5/2
1440 RETURN
1450 IF(T-I1)>(2*I)/1440 THEN 1470
1460 GOTO 1490
1470 PRINT
1480 IF LP$="Y" THEN LPRINT:LN=LN+1
1490 IF FNA(D1)<0 THEN 1560
1500 IF FNA(G)<G1 THEN 1560
1510 IF FNA(G)>G2 THEN 1530
1520 S$="U":GOTO 1570
1530 IF FNA(G)<G3 THEN S$="W":GOTO 1570
1540 IF FNA(G)>G4 THEN 1560
1550 S$="J":GOTO 1570
1560 S$=" "
1570 PRINT USING F2$;(INT(B+.5)),FNA(A),FNA(E),FNA(G),FNA(D1),S$
1580 IF LP$<>"Y" THEN 1610
1590 LPRINT USING F2$;(INT(B+.5)),FNA(A),FNA(E),FNA(G),FNA(D1),S$
1600 LN=LN+1: IF LN>=60 THEN GOSUB 1850
1610 I1=T
1620 B=B+I
1630 Z=(B-INT(B/100)*100)-60
1640 IF Z<0 THEN 720
1650 B=INT(B/100)*100+100+Z
1660 GOTO 720
1670 PRINT
1680 GOTO 1750
1690 PRINT "DO YOU WANT THIS DATA FOR ADDITIONAL DATES? (Y/N)"
1700 INPUT D$
1710 IF D$="N" THEN END
1720 PRINT:PRINT
1730 IF LP$="Y" THEN LPRINT:LPRINT:LN=LN+2
1740 GOTO 350
1750 N=N-1:IF N=0 THEN 1690
1760 IF D<W(M) THEN D=D+1:GOTO 1820
1770 IF M<>2 THEN 1800
1780 IF (Y/4-INT(Y/4))>0 THEN 1800
1790 IF D=28 THEN D=D+1:GOTO 1820
1800 D=1:IF M<12 THEN M=M+1:GOTO 1820
1810 M=1:Y=Y+1:GOTO 1820
1820 PRINT
1830 IF LP$="Y" THEN LPRINT:LPRINT:LN=LN+2
1840 GOTO 440
1850 FOR J=1 TO FF:LPRINT:NEXT J
1860 IF FF>10 THEN LN=4:FF=10:RETURN
1870 LPRINT F1$:LPRINT:LN=6
1880 RETURN
1890 END
```

MOONTRACKER (Apple)

```
10  REM  MOON TRACKING PROGRAM (APPLE)
20  REM  ADAPTED FROM WA1JXN'S PROGRAM BY KC6A AND N6NB.  FOR EASE OF
30  REM  DATA INPUT, PRINTOUT OF AN UNLIMITED NUMBER OF CONSECUTIVE
40  REM  DAYS REQUIRES ONLY ONE ENTRY DATE.  LATITUDE/LONGITUDE INPUT
50  REM  IS AUTOMATIC FOR PRIMARY USERS, AND PRINTER OUTPUT IS FULLY
60  REM  PAGINATED.
```

```
70 PR$ =  CHR$ (4) + "PR#1":SC$ =  CHR$ (4) + "PR#0"
72  REM  FOR 80-COL VIDEO OR PRINTOUT ADD TO LINE 70 AS YOUR SYSTEM
74  REM  REQUIRES (USUALLY CHR$(9)+"80N" TO INITIALIZE PRINTER AND
76  REM  CHR$(4)+"PR#3" FOR VIDEO BOARDS).  THEN ADD SPACES TO PRINT LINES.
80  DIM W(12): REM  CALENDAR INFORMATION (LEAP YEAR COMPUTED LATER)
90 W(1) = 31:W(2) = 28:W(3) = 31:W(4) = 30:W(5) = 31:W(6) = 30
100 W(7) = 31:W(8) = 31:W(9) = 30:W(10) = 31:W(11) = 30:W(12) = 31
110  DEF  FN A(X) =  INT (X * D5 * 10 + .5) / 10
120  DEF  FN B(X) = (X -  INT (X)) * P5
130 P5 = 2 * 3.14159
140 D5 = 360 / P5
150 R5 = P5 / 360:FF = 10
160 F1$ = " UTC     AZ        EL       GHA      DEC"
170  REM
180  HOME : PRINT  TAB( 9)"MOON TRACKING PROGRAM"
190  PRINT : PRINT : INPUT "WHAT IS YOUR CALL SIGN ";Q$
200  PRINT : INPUT "DO YOU WANT HARD COPY (Y/N) ";LP$
210  IF Q$ = "KC6A" THEN L5 = 33:U5 = 47.1:L6 = 118:U6 = 5.95: GOTO 250
220  REM  PUT YOUR CALL AND COORDINATES HERE
230  INPUT "WHAT IS THE LATITUDE IN DEGREES, MINUTES";L5,U5
240  INPUT "  --AND LONGITUDE IN DEGREES, MINUTES";L6,U6
250 X5 = L5:X6 = L6
260 L5 = (L5 + U5 / 60) * R5:L6 = (L6 + U6 / 60) * R5
270  PRINT
280  PRINT "WHAT IS THE DESIRED PRINTING INCREMENT"
285  INPUT "  IN MINUTES (1 TO 60)";I
290  PRINT : PRINT "MOON TRACKING DATA COMPUTED FOR ";Q$
300  PRINT : PRINT "LATITUDE: ";X5;"-";U5;" LONGITUDE: ";X6;"-";U6
310  PRINT : PRINT "IF YOU ONLY WANT A PRINTOUT FOR TIMES"
312  PRINT "  WHEN THE MOON IS NEAR THE HORIZON,"
320  INPUT "  ENTER 'H' (OTHERWISE <CR>";B$
330 I6 = 100
340  IF B$ = "H" THEN  PRINT : INPUT "BELOW WHAT ELEV. IN DEGREES DO YOU WANT PRIN
TOUT TO OCCUR ";I6
350  PRINT : PRINT "WHAT IS THE STARTING UTC DATE (MO,DAY,"
352  INPUT "  YR  --EXAMPLE: 7,4,1984)";M,D,Y
360  PRINT : INPUT "FOR HOW MANY DAYS DO YOU WANT A PRINTOUT ";N
365  HOME
370  IF LP$ = "Y" THEN  PRINT PR$;
380  PRINT : PRINT : PRINT
390  PRINT : PRINT  TAB( 9)"MOON TRACKING PROGRAM"
400  PRINT : PRINT  TAB( 8)"DATA COMPUTED FOR ";Q$
410  PRINT : PRINT  TAB( 6)"LAT: ";X5;"-";U5;" LONG: ";X6;"-";U6
420  PRINT : PRINT "WINDOWS:  U=EUROPE; W=W/VE; J=JA/VK/ZL"
430  PRINT : PRINT :LN = 14
440 E1 = 2400:B = 0: GOTO 500
450 Y1 = Y - ( INT (Y / 100) * 100)
500  IF LN >  = 56 THEN FF = 70 - LN: GOSUB 1850
510  PRINT : PRINT  TAB( 5)"POSITION OF MOON ON ";M;"/";D;"/";Y
520  PRINT : PRINT F1$:LN = LN + 4
530 I1 = 2
540  REM  CALCULATION OF JULIAN DATE FOLLOWS
550  IF M >  = 3 THEN 610
560  IF  INT ((Y - 1853) / 4) < I1 THEN C1 = 0: GOTO 580
570 C1 =  - 1
580 J1 = 365 * (Y - 1853) + D + 30 * (M + 9) +  INT ((M + 10) / 2)
590 J2 =  INT ((Y - 1853) / 4) + 1 + C1
600  GOTO 690
610  IF  INT ((Y - 1852) / 4) < I1 THEN C1 = 0: GOTO 630
620 C1 =  - 1
630  IF M = 9 THEN 660
640  IF M = I1 THEN 660
650 C2 = 0: GOTO 670
660 C2 = 1
670 J1 = 365 * (Y - 1852) + D + 30 * (M - 3) +  INT ((M - 2) / 2)
680 J2 =  INT ((Y - 1852) / 4) + C1 + C2
690 J = J1 + J2
700  REM  (JULIAN DATE-2397547.5) FOR 0 HOURS UTC
710 T1 = J - 17472.5
720 D9 = (B -  INT (B / 100) * 100) +  INT (B / 100) * 60
```

```
730 D6 = (E1 -  INT (E1 / 100) * 100) +  INT (E1 / 100) * 60
740 D7 = D9 - D6
750 D8 = D7 - I
760  IF D7 > 0 THEN 780
770  GOTO 800
780  IF D8 >  = 0 THEN 1670
790 B = E1
800 T = (B -  INT (B / 100) * 100) / 1440 +  INT (B / 100) / 24
810 T5 = T1 + T
820 K1 =  FN B(.751213 + .036601102 * T5)
830 K2 =  FN B(.822513 + .0362916457 * T5)
840 K3 =  FN B(.995766 + .00273777852 * T5)
850 K4 =  FN B(.974271 + .0338631922 * T5)
860 K5 =  FN B(.0312525 + .03674819 57 * T5)
870 L8 = K1 + .658 * R5 *  SIN (2 * K4) + 6.289 * R5 *  SIN (K2)
880 L8 = L8 - 1.274 * R5 *  SIN (2 * K4) - .186 * R5 *  SIN (K3)
890 L8 = L8 + .214 * R5 *  SIN (K2 - 2 * K2) - .114 * R5 *  SIN (2 * K5)
900 L8 = L8 - .059 * R5 *  SIN (2 * K2 - 2 * K4) - .057 * R5 *  SIN (K2 + K3 - 2 *
 K4)
910 K6 = K5 + .6593 * R5 *  SIN (2 * K4) + 6.2303 * R5 *  SIN (K2) - 1.272 * R5 *
 SIN (K2 - 2 * K4)
920 L7 = 5.144 * R5 *  SIN (K6) - .146 * R5 *  SIN (K5 - 2 * K4)
930 D1 =  COS (L7) *  SIN (L8) * .397821 +  SIN (L7) * .917463
940 D1 =  ATN (D1 / ( SQR (1 - D1 ^ 2)))
950 G1 = 50.5 + ((D1) / (.792)) * D5
960 G2 = 80 + ((D1) / (.808)) * D5)
970 G3 = 141.5 - ((D1) * (.738) * D5)
980 G4 = 170.5 - ((D1) * (.857) * D5)
990 A2 =  COS (L7) *  COS (L8) / COS (D1)
1000 A1 = ( COS (L7) *  SIN (L8) * .917463 -  SIN (L7) * .397821) /  COS (D1)
1010 A =  ATN (A1 / A2)
1020  REM  IF YOU TYPED ALL THIS WITHOUT MAKING ANY MISTAKES,
1030  REM  YOU BELONG IN A SECRETARIAL POOL!
1040  GOSUB 1300
1050 R1 = A
1060 L1 = .065709822 * T1
1070 L = T * 24 * 1.00274 + 6.64606 + (L1 -  INT (L1 / 24) * 24)
1080 L = (L -  INT (L / 24) * 24)
1090 G = (L / 24) * P5 - R1
1100  IF G < P5 THEN 1130
1110 G = G - P5
1120  GOTO 1160
1130  IF G < 0 THEN 1150
1140  GOTO 1160
1150 G = G + P5
1160 H = L6 - G
1170 E3 =  COS (L5) *  COS (H) *  COS (D1) +  SIN (D1) *  SIN (L5)
1180 E2 =  SQR (1 - (E3 * E3))
1190 E =  ATN (E3 / E2)
1200  IF E < 0 THEN 1620
1210  IF E > I6 * R5 THEN 1620
1220 A2 =  SIN (D1) / ( COS (L5) *  COS (E))
1230 A2 = A2 - ( SIN (L5) /  COS (L5)) * ( SIN (E) /  COS (E))
1240 A1 =  SIN (L5) *  SIN (D1) +  COS (L5) *  COS (D1) *  COS (H)
1250 A1 = ( SIN (H) *  COS (D1)) / ( SQR (1 - A1 ^ 2))
1260 A =  ATN (A1 / A2)
1270  GOSUB 1300
1280  GOTO 1450
1290  REM
1300  IF A = 0 THEN 1320
1310  GOTO 1360
1320  IF A2 < 0 THEN 1340
1330  GOTO 1440
1340 A = P5 / 2
1350  GOTO 1440
1360  IF A > 0 THEN 1420
1370  IF A2 < 0 THEN 1400
1380 A = P5 + A
1390  GOTO 1440
```

```
1400 A = P5 + (A - P5 / 2)
1410  GOTO 1440
1420  IF A2 > 0 THEN 1440
1430 A = A + P5 / 2
1440  RETURN
1450  IF (T - I1) > (2 * I) / 1440 THEN 1480
1460  GOTO 1490
1480  PRINT :LN = LN + 1
1490  IF  FN A(D1) < 0 THEN 1560
1500  IF  FN A(G) < G1 THEN 1560
1510  IF  FN A(G) > G2 THEN 1530
1520 S$ = "U": GOTO 1562
1530  IF  FN A(G) < G3 THEN S$ = "W": GOTO 1562
1540  IF  FN A(G) > G4 THEN 1560
1550 S$ = "J": GOTO 1562
1560 S$ = " "
1562 BI =  INT (B + .5)
1563 AZ =  FN A(A):EL =  FN A(E):GHA =  FN A(G):DEC =  FN A(D1)
1564 PL$ = "+": IF DEC < 0 THEN PL$ = ""
1565 AZ$ =  STR$ (AZ): IF AZ =  INT (AZ) THEN AZ$ =  STR$ (AZ) + ".0"
1566 EL$ =  STR$ (EL): IF EL =  INT (EL) THEN EL$ =  STR$ (EL) + ".0"
1567 GH$ =  STR$ (GHA): IF GHA =  INT (GHA) THEN GH$ =  STR$ (GHA) + ".0"
1568 DE$ = PL$ +  STR$ ( INT (DEC * 10 + .5) / 10) + S$
1569  IF DEC =  INT (DEC) THEN DE$ = PL$ +  STR$ (DEC) + ".0" + S$
1570  PRINT  SPC( 4 -  LEN ( STR$ (BI)));BI; SPC( 9 -  LEN (AZ$));AZ$; SPC( 8 -  L
EN (EL$));EL$; SPC( 9 -  LEN (GH$));GH$; SPC( 9 -  LEN (DE$));DE$
1600 LN = LN + 1: IF LN >  = 60 THEN  GOSUB 1850
1610 I1 = T
1620 B = B + I
1630 Z = (B -  INT (B / 100) * 100) - 60
1640  IF Z < 0 THEN 720
1650 B =  INT (B / 100) * 100 + 100 + Z
1660  GOTO 720
1670  PRINT :LN = LN + 1
1680  GOTO 1750
1690  PRINT SC$: PRINT "DO YOU WANT THIS DATA"
1700  INPUT "FOR ADDITIONAL DATES? (Y/N) ";D$
1710  IF D$ = "N" THEN  END
1720  PRINT "SET PRINTER TO THE TOP OF A NEW PAGE"
1722  INPUT "THEN ENTER <CR> ";TF$
1740  GOTO 350
1750 N = N - 1: IF N = 0 THEN 1690
1760  IF D < W(M) THEN D = D + 1: GOTO 1820
1770  IF M <  > 2 THEN 1800
1780  IF (Y / 4 -  INT (Y / 4)) > 0 THEN 1800
1790  IF D = 28 THEN D = D + 1: GOTO 1820
1800 D = 1: IF M < 12 THEN M = M + 1: GOTO 1820
1810 M = 1:Y = Y + 1
1820  PRINT : PRINT :LN = LN + 2
1840  GOTO 440
1850  FOR J = 1 TO FF: PRINT : NEXT J
1860  IF FF > 10 THEN LN = 4:FF = 10: RETURN
1870  PRINT F1$: PRINT :LN = 6
1880  RETURN
1890  END
```

MOONTRACKER (Commodore 64)

```
5 OPEN 130,2,2,CHR$(136):REM 1200B/2ST
10 REM MOON TRACKING PROGRAM (COMMODORE)
20 REM SEE OTHER VERSIONS FOR INITIAL CREDITS
30 REM
40 REM
50 REM
```

```
60 REM
70 PRINT CHR$(147):REM CLEAR SCREEN
80 DIM W(12)
90 W(1)=31:W(2)=28:W(3)=31:W(4)=30:W(5)=31:W(6)=30
100 W(7)=31:W(8)=31:W(9)=30:W(10)=31:W(11)=30:W(12)=31
110 DEF FNA(X)=INT(X*D5*10+.5)/10
120 DEF FNB(X)=(X-INT(X))*P5
130 P5=2*3.14159:D5=360/P5:R5=P5/360:FF=10
140 F1$="  UTC       AZ          EL       GHA      DEC"
180 PRINT:PRINT TAB(9)"MOON TRACKING PROGRAM"
185 PRINT:INPUT "WHAT IS YOUR CALL SIGN";Q$
190 PRINT:INPUT"DO YOU WANT A HARD COPY (Y/N)";LP$
210 IF Q$="N6NB" THEN L5=33:U5=48:L6=117:U6=50:GOTO 250
212 PRINT:PRINT "ENTER YOUR COORDINATES: "
215 PRINT "   USE '-' FOR BOTH DEGREES AND MINS"
217 PRINT "   IF SOUTH LATITUDE, EAST LONGITUDE"
220 PRINT:INPUT "LATITUDE (E.G. 33,30)";L5,U5
230 INPUT "LONGITUDE (E.G. 88,24)";L6,U6
250 X5=L5:X6=L6
260 L5=(L5+U5/60)*R5:L6=(L6+U6/60)*R5
270 PRINT:PRINT "WHAT IS THE DESIRED PRINTING INCREMENT"
280 INPUT "   IN MINUTES (1 TO 60)";I
290 PRINT:PRINT "MOON TRACKING DATA COMPUTED FOR ";Q$
300 PRINT:PRINT "LATITUDE: ";X5;"-";U5; "LONGITUDE: ";X6;"-";U6
310 PRINT:PRINT "IF YOU ONLY WANT A PRINTOUT FOR TIMES"
312 PRINT "  WHEN THE MOON IS NEAR THE HORIZON,"
320 INPUT "  ENTER 'H' (OTHERWISE <CR>)";B$
330 I6=100
340 IF B$="H" THEN INPUT "BELOW WHAT ELEV IN DEGREES DO YOU WANT A PRINTOUT";I6
350 PRINT:PRINT "WHAT IS THE STARTING GMT DATE?"
352 INPUT "ENTER THE MONTH (1-2 DIGITS)";M
353 INPUT "ENTER THE DAY (1-2 DIGITS)";D
354 INPUT "ENTER THE YR (4 DIGITS)";Y
360 PRINT:PRINT "FOR HOW MANY DAYS DO YOU WANT"
362 INPUT "A PRINTOUT";N
365 PRINT CHR$(147)
370 IF LP$="Y" THEN CMD 130
372 PRINT:PRINT:PRINT
380 PRINT:PRINT TAB(9)"MOON TRACKING PROGRAM"
400 PRINT:PRINT TAB(8)"DATA COMPUTED FOR ";Q$
410 PRINT:PRINT TAB(6)"LAT: ";X5;"-";U5;" LONG: ";X6;"-";U6
420 PRINT:PRINT "WINDOWS:  U=EUROPE; W=W/VE; J=JA/VK/ZL"
430 PRINT:PRINT:LN=14
440 E1=2400:B=0:GOTO 500
450 Y1=Y-(INT(Y/100)*100)
500 IF LN>=56 THEN FF=70-LN:GOSUB 1850
510 PRINT:PRINT "   POSITION OF MOON ON ";M;"/";D;"/";Y
520 PRINT:PRINT F1$:LN=LN+4
530 I1=2
540 REM CALCULATING JULIAN DATE
550 IF M>=3 THEN 610
560 IF INT((Y-1853)/4)<11 THEN C1=0:GOTO 580
570 C1=-1
580 J1=365*(Y-1853)+D+30*(M+9)+INT((M+10)/2)
590 J2=INT((Y-1853)/4)+1+C1
600 GOTO 690
610 IF INT((Y-1852)/4)<11 THEN C1=0:GOTO630
620 C1=-1
630 IF M=9 THEN 660
640 IF M=11 THEN 660
650 C2=0:GOTO 670
660 C2=1
670 J1=365*(Y-1852)+D+30*(M-3)+INT((M-2)/2)
680 J2=INT((Y-1852)/4)+C1+C2
690 J=J1+J2
700 REM JULIAN DATE-2397547.5 FOR 0 HRS GMT
710 T1=J-17472.5
720 D9=(B-INT(B/100)*100)+INT(B/100)*60
730 D6=(E1-INT(E1/100)*100)+INT(E1/100)*60
```

```
740 D7=D9-D6
750 D8=D7-I
760 IF D7>0 THEN 780
770 GOTO 800
780 IF D8=>0 THEN 1670
790 B=E1
800 T=(B-INT(B/100)*100)/1440+INT(B/100)/24
810 T5=T1+T
820 K1=FNB(.751213+.036601102*T5)
830 K2=FNB(.822513+.0362916457*T5)
840 K3=FNB(.995766+.00273777852*T5)
850 K4=FNB(.974271+.0338631922*T5)
860 K5=FNB(.0312525+.0367481957*T5)
870 L8=K1+.658*R5*SIN(2*K4)+6.289*R5*SIN(K2)
880 L8=L8-1.274*R5*SIN(K2-2*K4)-.186*R5*SIN(K3)
890 L8=L8+.214*R5*SIN(2*K2)-.114*R5*SIN(2*K5)
900 L8=L8-.059*R5*SIN(2*K2-2*K4)-.057*R5*SIN(K2+K3-2*K4)
910 K6=K5+.6593*R5*SIN(2*K4)+6.2303*R5*SIN(K2)-1.272*R5*SIN(K2-2*K4)
920 L7=5.144*R5*SIN(K6)-.146*R5*SIN(K5-2*K4)
930 D1=COS(L7)*SIN(L8)*.397821+SIN(L7)*.917463
940 D1=ATN(D1/(SQR(1-D1^2)))
950 G1=50.5+((D1)/(.792))*D5
960 G2=80+((D1)/(.808))*D5
970 G3=141.5-((D1)*(.738)*D5)
980 G4=170.5-((D1)*(.857)*D5)
990 A2=COS(L7)*COS(L8)/COS(D1)
1000 A1=(COS(L7)*SIN(L8)*.917463-SIN(L7)*.397821)/COS(D1)
1010 A=ATN(A1/A2)
1020 REM
1030 REM
1040 GOSUB 1300
1050 R1=A
1060 L1=.065709822*T1
1070 L=T*24*1.00274+6.64606+(L1-INT(L1/24)*24)
1080 L=(L-INT(L/24)*24)
1090 G=(L/24)*P5-R1
1100 IF G<P5 THEN 1130
1110 G=G-P5
1120 GOTO 1160
1130 IF G<0 THEN 1150
1140 GOTO 1160
1150 G=G+P5
1160 H=L6-G
1170 E3=COS(L5)*COS(H)*COS(D1)+SIN(D1)*SIN(L5)
1180 E2=SQR(1-(E3*E3))
1190 E=ATN(E3/E2)
1200 IF E<0 THEN 1620
1210 IF E>I6*R5 THEN 1620
1220 A2=SIN(D1)/(COS(L5)*COS(E))
1230 A2=A2-(SIN(L5)/COS(L5))*(SIN(E)/COS(E))
1240 A1=SIN(L5)*SIN(D1)+COS(L5)*COS(D1)*COS(H)
1250 A1=(SIN(H)*COS(D1))/(SQR(1-A1^2))
1260 A=ATN(A1/A2)
1270 GOSUB 1300
1280 GOTO 1450
1290 REM
1300 IF A=0 THEN 1320
1310 GOTO 1360
1320 IF A2<0 THEN 1340
1330 GOTO 1440
1340 A=P5/2
1350 GOTO 1440
1360 IF A>0 THEN 1420
1370 IF A2<0 THEN 1400
1380 A=P5+A
1390 GOTO 1440
1400 A=P5+(A-P5/2)
1410 GOTO 1440
1420 IF A2>0 THEN 1440
1430 A=A+P5/2
```

```
1440 RETURN
1450 IF (T-I1)>(2*I)/1440 THEN 1480
1460 GOTO 1490
1480 PRINT:LN=LN+1
1490 IF FNA(D1)<0 THEN 1560
1500 IF FNA(G)<G1 THEN 1560
1510 IF FNA(G)>G2 THEN 1530
1520 S$="U":GOTO 1562
1530 IF FNA(G)<G3 THEN S$="W":GOTO 1562
1540 IF FNA(G)>G4 THEN 1560
1550 S$="J":GOTO 1562
1560 S$=" "
1562 BI=INT(B+.5)
1563 AZ=FNA(A):EL=FNA(E):GHA=FNA(G):DEC=FNA(D1)
1564 PL$="+":IF DEC<0 THEN PL$=""
1565 AZ$=STR$(AZ):IF AZ=INT(AZ) THEN AZ$=STR$(AZ)+".0"
1566 EL$=STR$(EL):IF EL=INT(EL) THEN EL$=STR$(EL)+".0"
1567 GH$=STR$(GHA):IF GHA=INT(GHA) THEN GH$=STR$(GHA)+".0"
1568 DE$=PL$+STR$(INT(DEC*10+.5)/10)+S$
1569 IF DEC=INT(DEC)THEN DE$=PL$+STR$(DEC)+".0"+S$
1570 PRINT SPC(5-LEN(STR$(BI)));BI;SPC(8-LEN(AZ$));AZ$;SPC(7-LEN(EL$));EL$;
1572 PRINT SPC(9-LEN(GH$));GH$;SPC(9-LEN(DE$));DE$
1600 LN=LN+1:IF LN>=60 THEN GOSUB 1850
1610 I1=T
1620 B=B+I
1630 Z=(B-INT(B/100)*100)-60
1640 IF Z<0THEN 720
1650 B=INT(B/100)*100+100+Z
1660 GOTO 720
1670 PRINT:LN=LN+1
1680 GOTO 1750
1690 IF LP$="Y" THEN PRINT#130
1692 PRINT "DO YOU WANT THIS DATA"
1700 INPUT "FOR ADDITIONAL DATES? (Y/N) ";D$
1710 IF D$="N" THEN 1890
1720 PRINT "SET PRINTER TO THE TOP OF A NEW PAGE"
1722 INPUT "THEN ENTER <CR> ";TF$
1740 GOTO 350
1750 N=N-1:IF N=0 THEN 1690
1760 IF D<W(M) THEN D=D+1:GOTO 1820
1770 IF M<>2 THEN 1800
1780 IF (Y/4-INT(Y/4))>0 THEN 1800
1790 IF D=28 THEN D=D+1:GOTO 1820
1800 D=1:IF M<12 THEN M=M+1:GOTO 1820
1810 M=1:Y=Y+1
1820 PRINT:PRINT:LN=LN+2
1840 GOTO 440
1850 FOR J=1 TO FF:PRINT:NEXT J
1860 IF FF>10 THEN LN=4:FF=10:RETURN
1870 PRINT F$:PRINT:LN=6
1880 RETURN
1890 IF LP$<>"Y" THEN 1900
1892 PRINT#130:FOR J=1TO1000:PRINT;:NEXT J
1894 CLOSE 130
1900 END
```

[redacted] MOONTRACKER (Timex/Sinclair) [redacted]

```
10 REM MOON TRACKING PROGRAM (ZX81 VERSION)
15 PI=3.14159
20 LET P5=2*PI
30 LET D5=360/P5
40 LET R5=P5/360
50 PRINT "ENTER YOUR LATITUDE"
```

```
55 PRINT "DEGREES?"
60 INPUT L5
65 PRINT "MINUTES?"
70 INPUT U5
80 PRINT "ENTER YOUR LONGITUDE"
85 PRINT "DEGREES?"
90 INPUT L6
95 PRINT "MINUTES?"
100 INPUT U6
110 LET X5=L5+U5/60
120 LET X6=L6+U6/60
130 LET L5=X5*R5
140 LET L6=X6*R5
150 PRINT "ENTER DESIRED DATE"
155 PRINT "MONTH?"
160 INPUT M
165 PRINT "DAY?"
170 INPUT D
175 PRINT "YEAR (4 DIGITS)?"
180 INPUT Y
190 PRINT
200 PRINT "ENTER DESIRED PRINTING INCREMENT (E.G.30)"
210 INPUT I
220 PRINT "EME DATA FOR ";X5;" DEGR.LAT. AND ";X6;" DEGR.LONG"
230 PRINT "ON ";M;" ";D;" ";Y
240 LET I6=100
250 LET E1=2400
260 LET B=0
270 PRINT "GMT    AZ     EL    GHA    DECL"
280 LET I1=2
290 IF M>=3 THEN GOTO 370
300 IF INT((Y-1853)/4)<11 THEN GOTO 330
310 C1=-1
320 GOTO 340
330 LET C1=0
340 LET J1=365*(Y-1853)+D+30*(M+9)+INT((M+10)/2)
350 LET J2=INT((Y-1853)/4)+1+C1
360 GOTO 480
370 IF INT ((Y-1852)/4)<11 THEN GOTO 400
380 LET C1=-1
390 GOTO 410
400 LET C1=0
410 IF M=9 THEN GOTO 450
420 IF M=11 THEN GOTO 450
430 LET C2=0
440 GOTO 460
450 LET C2=1
460 LET J1=365*(Y-1852)+D+30*(M-3)+INT((M-2)/2)
470 LET J2=INT((Y-1852)/4)+C1+C2
480 LET J=J1+J2
490 PRINT
500 LET T1=J-17472.5
510 LET D9=(B-INT(B/100)*100)+INT(B/100)*60
520 LET D6=(E1-INT(E1/100)*100)+INT(E1/100)*60
530 LET D7=D9-D6
540 LET D8=D7-I
550 IF D7>0 THEN GOTO 570
560 GOTO 590
570 IF D8>0 THEN STOP
580 LET B=E1
590 LET T=(B-INT(B/100)*100)/1440+INT(B/100)/24
600 LET T5=T1+T
610 LET X=.751213+.036601102*T5
620 LET K1=(X-INT(X))*P5
630 LET X=.822513+.0362916457*T5
640 LET K2=(X-INT(X))*P5
650 LET X=.995766+.00273777852*T5
660 LET K3=(X-INT(X))*P5
670 LET X=.974271+.0338631922*T5
```

```
680 LET K4=(X-INT(X))*P5
690 LET X=.0312525+.03674819577*T5
700 LET K5=(X-INT(X))*P5
710 LET L8=K1+.658*R5*SIN(K2-2*K4)+6.289*R5*SIN(K2)
720 LET L8=L8-1.274*R5*SIN(2*K2)-.114*R5*SIN(2*K5)
730 LET L8=L8+.214*R5*SIN(2*K2-2*K4)-.057*R5*SIN(K2+K3-2*K4)
740 LET L8=L8-.059*R5*SIN(2*K4)+6.2303*R5*SIN(K2)-1.272*R5*SIN(K2-2*K4)
750 LET K6=K5+.6593*R5*SIN(K6)-.146*R5*SIN(K5-2*K4)
760 LET L7=5.144*R5*SIN(L8)*.397821+SIN(L7)*.917463
770 LET D1=COS(L7)*SIN(L8)*.397821+SIN(L7)*.917463
780 LET D1=ATN(D1/(SQR(1-D1**2)))
790 LET G1=50.5+((D1)/(.792))*D5
800 LET G2=80+((D1)/(.808))*D5
810 LET G3=141.5-((D1)*(.738)*D5)
820 LET G4=170.5-((D1)*(.857)*D5)
830 LET A2=COS(L7)*COS(L8)/COS(D1)
840 LET A1=(COS(L7)*SIN(L8)*.917463-SIN(L7)*.397821)/COS(D1)
850 LET A=ATN(A1/A2)
860 GOSUB 1120
870 LET R1=A
880 LET L1=.065709822*T1
890 LET L=T*24*1.00274+6.64606+(L1-INT(L1/24)*24)
900 LET L=(L-INT(L/24)*24)
910 LET G=(L/24)*P5-R1
920 IF G<P5 THEN GOTO 950
930 LET G=G-P5
940 GOTO 980
950 IF G<0 THEN GOTO 970
960 GOTO 980
970 LET G=G+P5
980 LET H=L6-G
990 LET E3=COS(L5)*COS(H)*COS(D1)+SIN(D1)*SIN(L5)
1000 LET E2=SQR(1-(E3**2))
1010 LET E=ATN(E3/E2)
1020 IF E<0 THEN 1460
1030 IF E>I6*R5 THEN 1460
1040 LET A2=SIN(D1)/(COS(L5)*COS(E))
1050 LET A2=A2-(SIN(L5)/COS(L5))*(SIN(E)/COS(E))
1060 LET A1=SIN(L5)*SIN(D1)+COS(L5)*COS(D1)*COS(H)
1070 LET A1=(SIN(H)*COS(D1))/(SQR(1-A1**2))
1080 LET A=ATN(A1/A2)
1090 GOSUB 1120
1100 GOTO 1270
1110 REM
1120 IF A=0 THEN GOTO 1140
1130 GOTO 1180
1140 IF A2<0 THEN GOTO 1160
1150 GOTO 1260
1160 LET A=P5/2
1170 GOTO 1260
1180 IF A>0 THEN GOTO 1240
1190 IF A2<0 THEN GOTO 1220
1200 LET A=P5+A
1210 GOTO 1260
1220 LET A=P5+(A-P5/2)
1230 GOTO 1260
1240 IF A2>0 THEN GOTO 1260
1250 A=A+P5/2
1260 RETURN
1270 IF(T-I1)>(2*I)/1440 THEN GOTO 1290
1280 GOTO 1300
1290 PRINT
1300 IF (INT(D1*D5*10+.5)/10)<0 THEN GOTO 1430
1310 IF (INT(G*D5*10+.5)/10)<G1 THEN GOTO 1430
1320 IF (INT(G*D5*10+.5)/10)>G2 THEN GOTO 1340
1330 GOTO 1370
1340 IF (INT(G*D5*10+.5)/10)<G3 THEN GOTO 1390
1350 IF (INT(G*D5*10+.5)/10)>G4 THEN GOTO 1430
1360 GOTO 1410
```

```
1370 LET S$="U"
1380 GOTO 1440
1390 LET S$="W"
1400 GOTO 1440
1410 LET S$="J"
1420 GOTO 1440
1430 LET S$=" "
1440 PRINT (INT(B+.5));"  ";(INT(A*D5*10+.5)/10);"  ";(INT(E*D5*10+.5)/10);
       "  ";(INT(G*D5*10+.5)/10);"  ";(INT(D1*D5*10+.5)/10);S$
1450 LET I1=T
1460 LET B=B+I
1470 IF B>2400 THEN STOP
1480 LET Z=(B-INT(B/100)*100)-60
1490 IF Z<0 THEN 510
1500 LET B=INT(B/100)*100+100+Z
1510 GOTO 510
```

THE SKYLOCATOR

What it does

The SKYLOCATOR is a short program that computes tracking data for a celestial body, if you know its right ascension and declination. The results will be displayed for whatever time interval you wish. This program will run on almost any computer, including a VIC-20 *without* any additional memory.

Setup instructions

The listings for Apple and for TRS-80 and CP/M will run on many computers without modification. For Commodore, use the Apple version. Then add a line at a beginning to open a channel to your printer and another at the end to close it (see the modifications listed under the MOONTRACKER program for examples). Replace line 250 with this line:

250 IF LP$<>"Y" THEN CMD 130

(Note: Use your command channel in place of 130.) Then replace line 1010 with this line:

1010 PRINT#130

(Note: Use your command channel # in place of 130.)

For the Sinclair, the Apple print commands in lines 250 and 1010 should be deleted. Then break each two-value input statement into separate lines. For instance, make line 140 into two lines as follows:

```
140 INPUT R1
142 INPUT M1
```

Operating notes

This program will provide elevation and azimuth data for any celestial object outside the solar system. But note that the time given is sidereal time, not solar time. In this "star time," a day is about four minutes shorter than a solar day. You can use an almanac to determine the difference between sidereal time and local time on any particular day of the year. Note that the data computed by the program is valid for any day of the year—what changes from day to day is the difference between sidereal time and local time.

This program has a number of amateur radio applications. For one, it can be used to find a particularly noisy celestial object such as Cassiopeia A or Sagittarius A for EME receiver and antenna testing and calibration. You can compare the noise levels from these sources to a cold sky to determine how well your system is working.

Modifications See above. The CP/M and TRS-80 version should work without modifications, except for a revision of the clear-screen command in line 22, on either a TRS-80 or CP/M system running Microsoft BASIC-80.

SKYLOCATOR (TRS-80 and CP/M)

```
10 REM  SKY LOCATING PROGRAM - GIVES AZ-EL TRACKING DATA
12 REM FOR A CELESTIAL BODY IN SIDEREAL TIME (TRS-80 VERSION)
20 REM BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
22 CLS:REM CLEAR SCREEN
25  A$="CELESTIAL BODY TRACKING PROGRAM": PRINT TAB(16)A$:PRINT
30  INPUT "DO YOU WANT A HARD COPY?";LP$
40 PI = 3.14159:RAD = 360 / (2 * PI)
50  PRINT "ENTER OBSERVER'S LATITUDE IN DEGREES,MINUTES"
60  INPUT L1,M1
110 L1 = (L1 + M1 / 60) / RAD
130  PRINT "ENTER OBJECT'S RIGHT ASCENSION IN HOURS, MINUTES"
140  INPUT R1,M1
160 R1 = (R1 + M1 / 60)
170  PRINT "ENTER DECLINATION IN DEGREES, MINUTES"
180  INPUT D1,M2
200 D1 = (D1 + M2 / 60) / RAD
210  PRINT "ENTER TIME INTERVAL IN MINUTES"
220  INPUT I
222  INPUT "WHAT IS THE OBJECT'S NAME:";C$
227 U$ = "     ST                EL              AZ"
228 V$ = "    ####              ##.#             ###.#"
230  PRINT
240  PRINT
250 IF LP$="Y" THEN GOSUB 1000
260 PRINT TAB(5)"ALL TIMES SHOWN BELOW ARE SIDEREAL TIME":PRINT
262 PRINT U$:PRINT
270  FOR J = 0 TO 60 * 24 STEP I
280 S =  INT (J / 60) * 100 + J - 60 *  INT (J / 60)
290 H = (J / 60 - R1) * 15
300  IF H < 0 THEN H = 360 + H
310 H = H / RAD
320 E3 =  COS (L1) *  COS (H) *  COS (D1) +  SIN (D1) *  SIN (L1)
330 E2 =  SQR ( ABS (1 - (E3 * E3)))
340 E =  ATN (E3 / E2)
350  IF E < 0 THEN 640
360 A2 =  SIN (D1) / ( COS (L1) *  COS (E))
370 A2 = A2 - ( SIN (L1) /  COS (L1)) * ( SIN (E) /  COS (E))
380 A1 =  SIN (L1) *  SIN (D1) +  COS (L1) *  COS (D1) *  COS (H)
390 A1 = ( SIN (H) *  COS (D1)) / ( SQR ( ABS (1 - A1 * A1)))
400 A =  ATN (A1 / A2)
410  IF A <  > 0 THEN 460
420  IF A2 < 0 THEN 440
430  GOTO 600
440 A = PI
450  GOTO 600
460  IF A > 0 THEN 520
470  IF A2 < 0 THEN 500
480 A = 2 * PI + A
490  GOTO 600
500 A = 2 * PI + (A - PI)
510  GOTO 600
520  IF A2 > 0 THEN 600
```

```
530 A = A + PI
600 A = 2 * PI - A
610 EE =  INT (E * RAD * 100 + .5) / 100
620 AA =  INT (A * RAD * 100 + .5) / 100
630 PRINT USING V$;S,EE,AA
635 IF LP$="Y" THEN LPRINT USING V$;S,EE,AA
640  NEXT J
650 END
1000 LPRINT TAB(10)A$:LPRINT:LPRINT
1002 LPRINT TAB(5)"AZ-EL (IN SIDEREAL TIME) FOR: ";C$: LPRINT
1004 LPRINT U$: LPRINT:RETURN
```

SKYLOCATOR (Apple)

```
10  REM  SKY LOCATING PROGRAM - GIVES AZ-EL FOR A CELESTIAL BODY IN SIDEREAL TIME
12  REM  APPLE VERSION
20  REM  BY KC6A AND N6NB; (C) 1983 BY HAYDEN BOOK CO.,INC.
22  HOME
25  PRINT  SPC( 4)"CELESTIAL BODY TRACKING PROGRAM": PRINT
30  INPUT "DO YOU WANT A HARD COPY?";LP$
32 D$ =  CHR$ (4) + "PR#1":SC$ =  CHR$ (4) + "PR#0'
40 PI = 3.14159:RAD = 360 / (2 * PI)
50  PRINT "ENTER OBSERVER'S LATITUDE IN DEGR,MINS"
60  INPUT L1,M1
110 L1 = (L1 + M1 / 60) / RAD
130  PRINT "ENTER OBJECT'S RIGHT ASCENSION IN HOURS, MINS"
140  INPUT R1,M1
160 R1 = (R1 + M1 / 60)
170  PRINT "ENTER DECLINATION IN DEGREES, MINS"
180  INPUT D1,M2
200 D1 = (D1 + M2 / 60) / RAD
210  PRINT "ENTER TIME INTERVAL IN MINUTES"
220  INPUT I
222  INPUT "WHAT IS THE OBJECT'S NAME:";C$
227 U$ = "  ST        EL        AZ"
230  PRINT
240  PRINT
250  IF LP$ = "Y" THEN  PRINT D$
255  PRINT : PRINT
260  PRINT "AZ-EL (SIDEREAL TIME) FOR: ";C$: PRINT
265  PRINT U$: PRINT
270  FOR J = 0 TO 60 * 24 STEP I
280 S =  INT (J / 60) * 100 + J - 60 *  INT (J / 60)
290 H = (J / 60 - R1) * 15
300  IF H < 0 THEN H = 360 + H
310 H = H / RAD
320 E3 =  COS (L1) *  COS (H) *  COS (D1) +  SIN (D1) *  SIN (L1)
330 E2 =  SQR ( ABS (1 - (E3 * E3)))
340 E =  ATN (E3 / E2)
350  IF E < 0 THEN 1000
360 A2 =  SIN (D1) / ( COS (L1) *  COS (E))
370 A2 = A2 - ( SIN (L1) /  COS (L1)) * ( SIN (E) /  COS (E))
380 A1 =  SIN (L1) *  SIN (D1) +  COS (L1) *  COS (D1) *  COS (H)
390 A1 = ( SIN (H) *  COS (D1)) / ( SQR ( ABS (1 - A1 * A1)))
400 A =  ATN (A1 / A2)
410  IF A <  > 0 THEN 460
420  IF A2 < 0 THEN 440
430  GOTO 600
440 A = PI
450  GOTO 600
460  IF A > 0 THEN 520
470  IF A2 < 0 THEN 500
480 A = 2 * PI + A
490  GOTO 600
500 A = 2 * PI + (A - PI)
510  GOTO 600
520  IF A2 > 0 THEN 600
```

```
530 A = A + PI
600 A = 2 * PI - A
610 EE =  INT (E * RAD * 100 + .5) / 100
620 AA =  INT (A * RAD * 100 + .5) / 100
630  PRINT  SPC( 5 -  LEN ( STR$ (S)));S; SPC( 10 -  LEN ( STR$ (EE)));EE; SPC( 10
 -  LEN ( STR$ (AA)));AA
1000  NEXT J
1010  PRINT SC$
1100  END
```

APPENDIX

IN CASE OF DIFFICULTY . . .

If you encounter difficulty with any of the programs in this book, these suggestions may be helpful.

1. Proofread carefully—Many of the difficulties you may experience with programs typed from published listings result from typing and proofreading errors. Even one missing comma, colon, or quotation mark—much less a missing line—can cause trouble.

If a program doesn't seem to work as it should, first check the accuracy of the program. If you have a printer, make a hard-copy listing of the program as you have typed it and carefully compare it with the original.

Make sure the line numbers are correct and in ascending order. A sure sign of a problem is finding a line out of order or a line that doesn't begin with a number. If you have typed the Apple version of the listings, note that any line longer than 80 characters in the Apple listings "wraps" to the next line without any indentation. Any line in the listing that begins with anything except a line number higher than the previous one is a continuation of the previous line. Be sure you typed it that way.

On the other hand, no line in any Commodore listing is supposed to be longer than 80 characters—that is the maximum allowable line length for Commodore. In a Commodore program listing, any line that "wraps" is a sign of trouble.

Perhaps an old newspaper editor's trick will help with the proofreading. Have someone *unfamiliar* with the program read what you

have typed aloud, including every last bit of punctuation. If you follow along by reading the listing in the book, the chances are good that you will spot any errors.

2. Make sure you have the right hardware for the program— Another common source of difficulty with programs such as these stems from hardware-software incompatibility. Be sure you typed the version of the program that was written for your system specifically, and not some other version. There is more than one listing of almost every program. The first or second line of each listing concludes by indicating the computer for which it is intended—except in the case of a few programs designed to run on several different systems.

Also, is your hardware (and operating system) like the system specified in the text? To find out if this might be a problem, reread the sections of Chapter 5 that pertain to your brand of equipment. Does your equipment match the description? Do you have the same printer interface? The same video display? Do you have an earlier or later model that may differ in some inconspicuous but important way?

Programming for Apple and Commodore computers presents a special problem because the syntax for hard-copy printouts may vary with variations in hardware. The programs assume that the Apple printer interface will be in slot #1. Even if the slot placement is correct, some printer interfaces may not respond correctly to certain commands in the programs. And for Commodore, the printer channel itself must be changed if the type of printer interface is changed. The programs in this book use the RS-232 channel for printing. If your system doesn't use that channel, or if the baud rate or handshaking differs from ours, you will need to adapt the syntax in each Commodore program for your own system.

If you're having problems with the video display, or if the hard-copy printout doesn't do tabbing or pagination correctly, there may be a hardware incompatibility problem. For example, many of these programs are written to skip over the perforations on fan-folded forms. If your printer or printer interface does this automatically, you may get double perforation-skipping (the result: lots of white space and bad pagination).

These programs will work properly on a wide variety of hardware, but in many cases modifications will be necessary if the hardware or operating system differs from that for which the program was written.

However, before you begin considering major surgery, make sure you haven't overlooked the obvious. For instance, is the printer turned on? Is its interconnecting cable plugged into the computer?

3. Be sure you're following the operating instructions— Some of the programs in this book are quite sophisticated and necessarily complex. There are many menu options in some programs; you should make sure you understand what each command does. Computer dealers say that 90 percent of their requests for servicing stem not from equipment failures but from someone missing a crucial step in the operating procedures. Because of their interest in hardware, most radio amateurs are less prone to cockpit troubles than are less technically aware computer owners. But

it's still possible there might be some detail you could have overlooked. Double check what the text (and the menus) say about the program.

4. Do you have the correct disk in the correct drive?—This is obvious to most readers. But it's possible to have problems because a disk is in the wrong disk drive, is full, or is write-protected. Many of the programs involve reading and writing data files on disk. Unless you change the syntax, the programs assume that the data disk will be in the first (or logged) disk drive. The data disk must have plenty of free space for data files, and it must not be write-protected. Only program disks—not data disks—should be write-protected.

Also, if you purchased the disk version of these programs instead of typing them from listings, remember that the distribution diskette is completely full. Its purpose is to get the programs from us to you; it is not intended to be a working disk. Each major program should be loaded and then saved on another disk (a newly formatted one). You'll need plenty of free disk space for the data files created by the data base programs, including the awards, logging and contest logging programs.

On the other hand, if you're running a program that *reads* a data file, be sure the proper file is on the disk in the logged drive. Most of the programs in Chapter 7, for example, must correctly read the latitude-longitude data file to run properly.

5. "Just like you said in QST except . . ."—Almost everyone who has ever written a construction article for an amateur radio magazine has had this experience: someone writes and says, "Your gizmo doesn't work. I built it just like you said in QST, except . . ." Then the letter-writer off-handedly says he changed something, and it turns out to be something that was crucial to the gizmo's proper operation. All kinds of unpredictable things can happen when a program is revised. If you're an experienced programmer, please forgive us for suggesting this, but in some cases modifications can cause problems. Our suggestion: get the programs running the way they're written first, save them on disk or cassette, and *then* make the modifications.

6. Would a ready-to-run program disk be worthwhile?—Some of the programs in this book require a fair amount of typing. They wouldn't do all of the things they do if they were much shorter. Because typing programs isn't everyone's favorite sport—and proofreading really isn't anyone's idea of fun—these programs are available on floppy disks for Apple II and other computers.

INDEX

ALTAIR 8800, *13*
ANTENNA MATCHING EVALUATOR (program), *7-8, 291-293*
ANTENNA SCALER (program), *7, 272-284*
Antennas (theory and design), *271-273*
Apple Computer, Inc., *15, 17-19*
Apple II, *1, 8, 14-15, 17-18, 22-25, 39, 41-43, 46-48, 52-53, 61-66, 69-71, 74*
Applesoft BASIC, *47-48, 55, 61-66, 69-71, 74*
Atari computers, *16-17, 22-24*
AWARDS DATA BASE (programs), *2-3, 76-77, 96-118*
AWRDINFOSETUP (program), *97, 118-120*
BASIC (language), *54-75*
  difference on various computers, *61-75*
  disk access commands, *61-64*
  hard-copy printout, *64-66*
BEAMHEADING CHART (program), *4, 132-139*
Clocks (real time), *73, 196-197*
Commodore 64, *1, 17, 22-23, 39, 41, 46-48, 67-69, 74*
Commodore VIC-20, *1, 17, 22-23, 67-69*
Compilers (BASIC), *47-49*
CONTEST LOGGERS (program), *5-7, 48, 195-264*
  FIELD DAY LOGGER, *6-7, 195-198, 221-249*
  GENERAL CONTEST LOGGER, *5-6, 195-236*
  SWEEPSTAKES LOGGER, *7, 195-198, 221, 249-264*
CP/M (operating system), *1, 15, 17, 28-29, 51-53, 55*
Digital Equipment Company, *12, 18*
Disks, floppy, *29-33, 50-52*
DUPECHECKER (program), *5, 48, 171-186, 270*
DUPEPRINT (program), *5, 186-195*
DX CHECKER (program), *4-5, 168-169*
DX DISPLAY (program), *4, 139-145*
EME SYSTEM ANALYZER (program), *8, 295-306*
ENIAC (computer), *10-11, 13*
Flowchart (CONTEST LOGGER), *223-226*
Flowchart (MINI-LOGGER), *57*
GRAYLINE (program), *4, 48, 151-163*
GRIDLOCATOR (program), *3, 76-77, 120-122*
Heath H-89, *14*
Hewlett-Packard, *19*
IBM Corporation, *11-13, 31*
IBM PC, *1, 18, 22-23, 48, 55, 72-74*
Interference (RFI, EMI), *38-39, 49*
Japanese computer manufacturers, *18-19, 31-32*
Jobs, Steven, *17, 19*
Kaypro computers, *71*
LATLONG (data file), *3-4, 124-132*
LOGBOOK (program), *2, 76-96*
LOGPRINT (program), *7, 265-270*
Microprocessors (CPUs), *13-14, 16, 22-23, 47-48*
Microsoft BASIC, *1, 48, 55, 61-66, 71-75*
MINILOGGER (program), *56-61, 76*
Modifying programs for various computers, *61-75*
  for Apple II, *69-71*
  for Commodore, *61, 67-69*
  for CP/M computers, *71-72*
  for IBM PC, *72-74*
  for TRS-80, *66-67*
MOONTRACKER (program), *8, 47-48, 295, 306-321*
MS-DOS (operating system), *18*
Northstar Horizon, *15, 24, 41, 48*
Operating techniques (in contests), *170-173, 221-222, 227-233*
Osborne, Adam, *17*
Parallel interfaces, *35-36*
PHASED VERTICAL PATTERN PLOTTER (program), *8, 284-291*
Printers, *34-35, 64-66*
Proofreading (for program errors), *325-326*
Random Access Memory (RAM), *25-29, 46-47*
Read Only Memory (ROM), *25-29*
S-100 bus, *13-14, 24*
Serial (RS-232-C) interfaces, *35-38*
SKYLOCATOR (program), *8, 295, 321-324*
SUNRISE CHART (program), *4, 145-150*
SUNRISES ANYWHERE (program), *5, 163-167*
Texas Instruments, *19*
Timex/Sinclair ZX-81, *8, 16, 22-23, 41, 48, 55, 318-321*
Troubleshooting hints, *61-75, 325-327*
  for Apple II, *69-71*
  for Commodore, *61, 67-69*
  for CP/M computers, *71-72*
  for IBM PC, *72-74*
  for TRS-80, *66-67*
TRS-80, *1, 14-16, 22-23, 25, 29, 41-43, 48, 50-53, 61-67, 74*
UNIVAC I (computer), *11*
Watson, Thomas, *11*
Xerox 820, *18, 24, 41, 48, 71*

0657-8

Computer Programs for Amateur Radio

Wayne Overbeck and James A. Steffen

The most comprehensive collection of BASIC programs ever published to help radio amateurs cut down on time spent keeping records, doing mathematical calculations, and beaming in on other stations.

Written by two radio amateurs who have nearly 50 years of combined experience and have received numerous awards, the book will show you how to use your microcomputer for contest logging, VHF operating, antenna design improvement, tracking grid locators and moonbounce communications, plus data base management programs to keep records for five-band DX Century Club, Worked All States, and general-purpose logging.

There's a program that will actually search out 1500 calls for duplicate stations in about two seconds! In addition, there are programs, using a data base file, with geographic coordinates for countries and cities all over the world which produce customized beamheading and distance charts, sunrise/sunset tables, and "grayline" calculations.

Other Books of Interest...

Basic Apple™ BASIC
James S. Coan

This book is a complete guide to Applesoft BASIC and takes the reader from beginning concepts, such as entering data and obtaining output, and planning programs, to more advanced topics, such as numeric and string arrays and sequential and random access fields. Alternative techniques for programming in Apple Integer BASIC are also provided. Both low-resolution and high-resolution graphics are discussed. #5226-5, paper, 256 pages.

Data Base Management for the Apple™
Nat Wadsworth

Learn the basics of storing and organizing information on your Apple II Plus or IIe home computer. This clearly written book includes DATA BASE, a simple, functional, and cross-referenced data base management program written in Applesoft BASIC. Using DATA BASE as a reference, the author shows how data base management techniques apply to common chores. #6282-6, paper, 128 pages.

HAYDEN BOOK COMPANY
a division of Hayden Publishing Company, Inc.
Hasbrouck Heights, New Jersey

ISBN 0-8104-0657-8